

Aplicació per monitoritzar la Qualitat del Servei

Aplicació per monitoritzar la
Qualitat del Servei



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



Autor: Rubén Soto Oseira
Director: Xavier Franch Gutiérrez - ESSI
Co-director: Marc Oriol Hilari - ESSI

Resum

El meu treball de fi de Grau és un projecte que està inclòs en un altre projecte més gran, anomenat SUPERSEDE, que pretén crear un sistema de millora d'aplicacions. Aquest sistema consistirà en un cicle amb diverses etapes, monitorització, anàlisi de dades i millora d'aplicacions a partir de l'anterior anàlisi. Com la pròpia definició de cicle indica, al acabar aquest, el que es pretén és que es pugui tornar a començar, de manera que les millores siguin continuades.

El meu projecte s'engloba en la part de monitorització, més concretament, en la monitorització d'aplicacions de mòbil.

A més, el nostre sistema, el que he hagut de desenvolupar en el projecte, consisteix en un servidor que et permet enviar una determinada configuració a aplicacions mòbils. Aquesta configuració és un document amb les diferents opcions de monitorització del sistema, com per exemple les diferents mètriques, l'interval de temps, el tòpic del servidor on es guardarà, etc. Aquestes dades, que han sigut demanades amb la configuració, són obtingudes a través de l'aplicació mòbil. Més tard, aquestes dades són enviades a altre servidor perquè siguin emmagatzemades per ser analitzades, en la segona fase del cicle de millora.

Resumen

Mi trabajo de fin de Grado es un proyecto que está incluido en otro proyecto más grande, llamado SUPERSEDE, que pretende crear un sistema de mejora de aplicaciones. Este sistema consistirá en un ciclo con diversas etapas, monitorización, análisis de datos y mejora de las aplicaciones a partir del anterior análisis. Como la propia definición de ciclo indica, al acabar éste, lo que se pretende es que se pueda volver a empezar, de manera que las mejoras sean continuadas.

Mi proyecto se engloba en la parte de monitorización, más concretamente, en la monitorización de aplicaciones de móvil.

Además, nuestro sistema, el cual he tenido que desarrollar durante el proyecto, consiste en un servidor que te permite enviar una determinada configuración a aplicaciones móviles. Esta configuración es un documento con las diferentes opciones de monitorización del sistema, como por ejemplo las diferentes métricas, el intervalo de tiempo, el tópico del servidor donde se va a guardar, etc. Estos datos, que han sido pedidos a través de la configuración, serán obtenidos a través de la aplicación móvil. Más tarde, estos datos son enviados a otro servidor para que sean almacenados para que sean analizados en la segunda fase del ciclo de mejora.

Abstract

My bachelor's thesis is a project included in another bigger project, called SUPERSEDE, that pretends to create a system of applications' improvements. This system will consist of a cycle of several steps, monitoring, data analysis and applications' improvement based on the previous analysis. As the cycle's definition indicates, at the end of it, we want to start the process again to be able to have continuous improvements.

My project is included in the monitoring phase. More concretely, it is part of the mobile applications' monitoring.

Besides, the system, which I have implemented during our thesis, consists in a server that allows to send a certain configuration to mobile applications. This configuration is a document where we can find the different monitoring options of the system, such as the different metrics, the time interval, the server's topic where it is going to be saved, etc. These data, which have been requested through the configuration document, are obtained in the mobile application. Afterwards, these data is sent to another server to be stored and later on, analysed in the second phase of the improvement cycle

Índex

1. Contextualització	6
a. Introducció	6
b. Actors implicats	6
2. Objectius.....	7
3. Estat de l'art	8
a. Contextualització.....	8
b. Explicació del problema.....	8
c. Limitacions d'altres projectes	8
d. Conclusions	9
4. Abast i possibles obstacles	10
a. Abast.....	10
b. Possibles obstacles	10
5. Metodologia i rigor	11
a. Mètodes de treball.....	11
b. Eines de seguiment	11
c. Mètodes de validació	11
6. Requisits funcionals	12
a. Casos d'ús	12
7. Requisits no funcionals	15
8. Esquema conceptual.....	18
9. Disseny	19
a. Arquitectura lògica.....	19
b. Arquitectura física	20
c. Servidor RESTful de Configuració.....	20
Diagrama de classes.....	21
Diagrames de seqüència	21
d. Aplicació de mòbil	23
Diagrama de classes.....	23
Diagrames de seqüència	24
e. Servidor RESTful d'enviament a Kafka	28
Diagrama de classes.....	28
Diagrames de seqüència	29

10. Implementació.....	30
a. Tecnologies utilitzades	30
i. Google Cloud Messaging.....	30
ii. Retrofit	30
iii. Java	31
iv. Android	31
b. Configuració de l'usuari	31
c. Funcionalitats desenvolupades.....	32
11. Validació	37
a. Entorn de proves	37
b. Pla de proves.....	37
c. Informe de proves	38
12. Planificació.....	40
a. Duració estimada prèvia del projecte	40
b. Consideracions prèvies	40
c. Fases del projecte.....	40
i. Fase de gestió del projecte	40
ii. Fase de desenvolupament.....	40
Servidor per les configuracions	40
Aplicació per realitzar la monitorització	40
Servidor per enviar les dades extretes	40
iii. Fase de documentació	41
d. Temps estimats	41
Temps estimats per fase	41
Temps estimats per fase i tipus de membre del projecte	41
e. Diagrama de Gant	42
f. Pla d'acció	42
g. Desviacions ocorregudes	43
13. Anàlisi de costos	45
a. Consideracions	45
b. Recursos utilitzats	45
Hardware.....	45
Software	45

Humans	45
c. Pressupost dels recursos humans	46
d. Pressupost pel hardware	46
e. Pressupost pel software	47
f. Contingència	47
g. Imprevistos	47
h. Control de gestió	48
i. Resum	48
14. Sostenibilitat i Compromís social	49
a. Econòmica	49
b. Social	49
c. Ambiental	49
d. Resum	50
15. Justificació de les competències tècniques	51
a. Coneixement d'assignatures d'especialitats	51
b. Característiques de l'especialitat	51
c. Competències tècniques	52
16. Conclusions	53
a. Assoliment d'objectius	53
b. Treball futur	53
c. Valoració personal	54
17. Referències	55

1. Contextualització

a. Introducció

El projecte que realitzarem tractarà de la monitorització d'una aplicació de mòbil. Aquest projecte es troba dins d'un altre projecte més gran anomenat SUPERSEDE.

SUPERSEDE consisteix en un gran projecte de la Unió Europea, amb la col·laboració de diverses universitats europees i diverses empreses importants en el camp de la informàtica, per dur a terme un projecte de monitorització de aplicacions, anàlisis de les dades i posterior millora d'aquestes.

Al tenir un temps limitat i ser el projecte tant gran, el meu projecte es basarà en aquesta primera part de recollida de dades, més concretament, en la monitorització de les aplicacions a millorar i l'enviament de les dades extretes pel posterior anàlisis.

b. Actors implicats

Els diferents actors implicats en aquest projecte són:

- La Universitat Politècnica de Catalunya (UPC) com a responsable de la monitorització de les aplicacions, tant l'extracció de la informació com l'enviament de les dades.
- La Fachhochschule Nordwestschweiz (FHNW) i la Universitat de Zúrich (UZH) com a responsables de la recollida de feedback dels usuaris. Aquests actors no estan directament involucrats amb la nostra part de monitorització però els afecten les nostres decisions d'arquitectura.
- La Universitat Politècnica de Catalunya (UPC) i Fondazione Bruno Kessler (FBK) són els coordinadors del projecte. En particular la UPC és la coordinadora científica. Aquests actors són els que estableixen els terminis i objectius del projecte a desenvolupar.
- La Universitat Politècnica de Catalunya (UPC) com a analista de les dades extretes en el procés de monitorització. Aprofitaran les dades per millorar posteriorment la aplicació monitoritzada.
- SENERCON, Siemens i Atos són les empreses on s'aplicaran els casos d'ús del projecte. Són els actors que decideixen quina informació necessiten sobre una determinada aplicació.

Com es pot comprovar, la UPC apareix tres vegades com a actor. Això és degut a què encara que sigui tot dins la UPC són grups d'investigació diferents els que duen a terme les diferents parts del projecte, per això he volgut separar-los en tres actors diferents.

2. Objectius

En el projecte podríem determinar com a dos, el número d'objectius principals del nostre projecte. Aquest dos objectius serien:

- Objectiu 1: Proveir d'un sistema de monitorització per a dispositius mòbils per poder obtenir la informació que desitgem. Aquest sistema ens ha de permetre personalitzar la configuració perquè es pugui adequar a les exigències i ens ha de permetre modificar aquesta configuració en tot moment.
- Objectiu 2: Proveir d'una infraestructura capaç d'unificar tota la informació extreta dels diferents dispositius monitoritzats per posteriorment ser analitzada.

3. Estat de l'art

a. Contextualització

La monitorització pot ser definida com el procés de recollida i enviament de informació rellevant sobre l'execució i evolució d'un sistema[1].

El sistema de software s'estan tornant més penetrants i incrustats a les nostres vides. Per aquesta raó, no és suficient la monitorització d'un sistema de software com a una entitat única i solitària. Necessitem considerar la informació contextual rellevant que pot ser recollida a través dels diferents sensors. D'altra banda, les monitoritzacions exhaustives necessiten anar més enllà de les fonts d'informació tècniques, necessiten considerar altres fonts d'informació, com pot ser la online per l'obtenció d'informació de l'usuari. Tota aquesta informació nova necessita ser afegida, també assegurant-ne la correctesa d'aquesta.

b. Explicació del problema

La monitorització ha sigut àmpliament estudiada en la literatura per garantir la correctesa d'una aplicació[2], dur un control de l'ús i comportament dels usuaris [3], i assegurar la qualitat del software en termes d'alguns factors de qualitat ja predefinits, per exemple, seguretat, l'eficiència, etc[4].

La monitorització de software ha evolucionat des de serveis de monitorització simple fins a serveis de monitorització distribuïts complexos[1]. La Internet de les coses ens permet monitoritzar el nostre voltant utilitzant sensor autònoms[5].

Quan monitoritzem la informació que ens arriba dels nostres consumidors de l'aplicació de software, distingim entre la informació implícita i explícita. En referència a la informació explícita, existeixen diverses propostes que recolzen la recollida de les opinions dels usuaris mitjançant la monitorització de les xarxes socials. Algunes enfocaments també recolzen el subseqüent anàlisi i adaptació d'aquesta informació[6]. En referència a l'altre tipus d'informació, la implícita, hi ha diverses propostes per monitoritzar l'ús d'una aplicació per identificar els patrons i posteriorment realitzar la anàlisi d'aquest[7].

c. Limitacions d'altres projectes

Hi ha diferents monitors de software per a dispositius mòbils i sensors per tasques específiques[15,16,17,18,19] però, actualment, aquests monitors són dirigits i operats sols, sense cap altre sistema que els recolzi, o estant limitats a un conjunt de mètriques específiques. Per el nostre millor coneixement, no hi ha enfocaments que proveeixin d'una solució de monitorització exhaustiva que permeti dirigir diversos monitors amb diferents propòsits, per exemple poder connectar i desconectar aquests monitors, poder configurar-los en temps d'execució, poder tenir un ampli

ventall de mètriques, com per exemple, CPU, memòria, capacitat del disc, utilització de la xarxa, la bateria, etc. Els diversos projectes que podem trobar actualment en el mercat[15,16,17,18,19] només contenen alguna d'aquestes característiques que té el conjunt del nostre sistema, fent d'aquest un millor producte respecte a d'altres sistemes amb un propòsit similar.

En el camp de l'adquisició de la informació, com indiquen Pagano i Bruegge[8], la falta de processos d'enginyeria condueix a una pobra col·lecció de mecanismes de recollida d'informació fent que penalitzi la qualitat de la informació recollida.

d. Conclusions

SUPERSEDE prova de corregir les manques dels anteriors sistemes proveint de diverses recerques. La primera contribució és l'aportació de mètodes i eines per dirigir tots els monitors i sensors en un ambient de monitorització integrat. La segona contribució, aporta tècniques per l'extracció d'informació implícita i explícita dels consumidors de les aplicacions a través de fonts online.

4. Abast i possibles obstacles

a. Abast

La implementació del projecte em durà a dues implementacions diferenciades que em donaran diferents utilitats.

La primera és el servidor RESTful que enviarà la configuració a la nostra aplicació. Aquesta part del sistema, ens permetrà rebre la configuració que volem monitoritzar. Cada vegada que vulguem canviar la configuració, només haurem d'enviar aquesta configuració al servidor RESTful, aquest s'encarregarà de enviar la configuració a tots els mòbils amb l'aplicació.

La segona és l'aplicació mòbil, que ens permetrà realitzar la monitorització de les aplicacions que hàgim posat a la configuració i les diferents mètriques que volem. Aquesta aplicació només estarà disponible sobre dispositius Android. L'aplicació ens permetrà dinàmicament modificar la configuració que hi hagi en aquell moment per una altra. A més tota la informació que hàgim recopilat anirà enviant-la a una altre servidor perquè els analistes puguin utilitzar aquesta informació posteriorment.

b. Possibles obstacles

Durant l'execució del projecte puc trobar-me algunes dificultats. Aquestes poden ser:

- Les restriccions temporals són un dels gran obstacles que em trobaré, ja que hi ha una limitació de temps molt gran per realitzar el projecte. Llavors qualsevol problema que pugui trobar durant la realització del projecte pot suposar un gran cost .
- Les tecnologies utilitzades en la implementació del projecte són conegudes però per la realització d'aquest projecte es necessiten determinats coneixements específics d'aquests llenguatges que s'ha d'anar aprenent alhora que programes. Per exemple, la implementació de REST sobre Java o el Google Cloud Messaging per Android i Java.
- Un altre obstacle que trobaré i que ja he esmentat anteriorment, és la plataforma mòbil sobre la que programaré. Degut al temps, l'aprenentatge que hauria de dur a terme i la infraestructura necessària, he decidit que l'aplicació només es farà per les aplicacions que sigui de la plataforma mòbil Android. Això fa que l'aplicació no valgui per tots els mòbils del mercat.
- L'accés a internet és un gran problema ja que la comunicació entre servidors i aplicació mòbil es realitza a través d'Internet. Llavors la connexió necessitem que sempre estigui disponible perquè el nostre sistema funcioni correctament.

5. Metodologia i rigor

a. Mètodes de treball

El mètode de treball utilitzat és Agile, ja que em permet molta més flexibilitat en els canvis. Cosa que em permetrà afegir canvis, si són necessaris, sense necessitat de tornar a començar pràcticament de nou. Aquesta metodologia em permet centrar-nos una mica més en el codi, conjuntament amb els seus artefactes d'especificació, disseny i test, ja que la documentació prèvia del projecte (Visió general, Arquitectura d'alt nivell, etc.) ja ha estat prèviament realitzada [20].

b. Eines de seguiment

La principal eina de seguiment del treball realitzat són les reunions setmanals amb el professor responsable del projecte, Marc Oriol, on es repassa tot el realitzat durant aquella setmana, es presenten els problemes que han aparegut i es fixa una possible ruta per la setmana vinent. A més, utilitzo l'eina Git pel control de versions en la implementació per poder anar seguint el progrés en el codi realitzat i poder solucionar més fàcilment qualsevol inconvenient que pugui aparèixer.

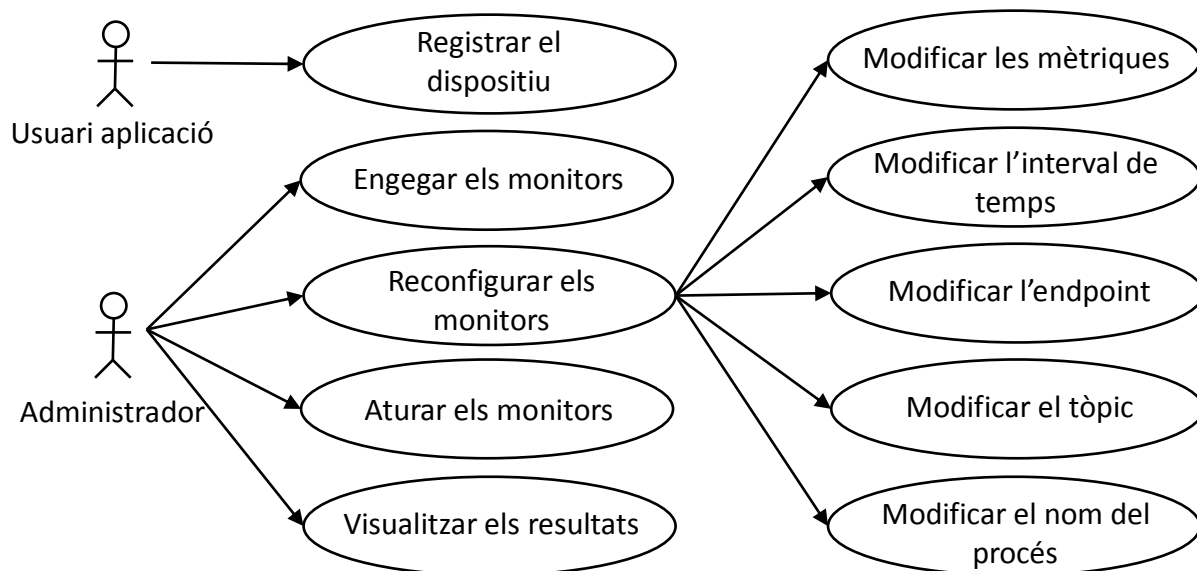
c. Mètodes de validació

La validació es realitza en un entorn de laboratori dins del context del projecte europeu, en particular en els casos d'ús de les empreses afectades pel projecte que hem esmentat abans. Però al ser un treball que té un límit de temps molt estricte, aquestes proves no podran ser realitzades durant la realització del projecte, ja que no es produiran fins passat un parell o tres de mesos després d'haver entregat el projecte. Per això, també es realitzarà un mètode de validació cada reunió amb el professor, per poder verificar que el treball realitzat va per el camí que desitgem.

6. Requisits funcionals

a. Casos d'ús

Els casos d'ús representen les accions que poden realitzar els usuaris en sistema. L'usuari només pot realitzar tres accions i tenim dos tipus d'usuaris, l'administrador i l'usuari de l'aplicació. Els casos d'ús del sistema són els següents:



Com podem apreciar en la figura anterior, l'usuari té poques funcionalitats en el sistema, fent d'aquest fet un avantatge per l'usuari. Ja que la seva interacció amb el sistema serà senzilla i fàcil d'aprendre.

A continuació, explicaré amb més detall cada cas d'ús mostrats en la figura anterior.

CAS D'ÚS #1: Registrar el dispositiu

Actor	Usuari que té l'aplicació instal·lada
Activador	Quan l'aplicació és executada per primera vegada
Precondicions	L'usuari ha de tenir l'aplicació instal·lada i tenir internet al dispositiu.
Escenari principal	<ol style="list-style-type: none">L'usuari executa l'aplicació.L'aplicació envia el seu marcatge de registre al servidor.El servidor se'l guarda si no el tenia i envia una resposta conforme ha arribat correctament el registre.
Extensions	En cas de que hi hagi algun error, el servidor no farà el registre i ensenyarà el codi d'error

CAS D'ÚS #2: Engegar els monitors

<i>Actor</i>	Administrador del sistema
<i>Activador</i>	L'administrador desitja enviar una monitorització amb una configuració determinada a tots els dispositius registrats.
<i>Precondicions</i>	Els servidors han d'estar en marxa
<i>Escenari principal</i>	<ol style="list-style-type: none">I. L'administrador indica la configuració que vol monitoritzar en els dispositius, tipus de mètriques, timeslot, endpoint, tòpic i nom del procés.II. Envia la informació als dispositiusIII. Aquests envien la informació demanada cada interval de temps determinat per la configuracióIV. Aquestes dades són desades en el servidor Kafka
<i>Extensions</i>	<p>En cas que l'administrador s'equivoqui en la configuració, l'aplicació retornarà un missatge d'error</p> <p>En cas que la connexió no pugui ser establerta entre l'aplicació i el servidor, retornarà un missatge d'error i no es realitzarà l'enviament.</p>

CAS D'ÚS #3: Reconfigurar els monitors

<i>Actor</i>	Administrador del sistema
<i>Activador</i>	L'administrador desitja enviar una monitorització amb una nova configuració per modificar la existent
<i>Precondicions</i>	Hi ha una configuració prèvia als dispositius
<i>Escenari principal</i>	<ol style="list-style-type: none">V. L'administrador indica la nova configuració que vol monitoritzar en els dispositius, tipus de mètriques, timeslot, endpoint, tòpic i nom del procésVI. Envia la informació als dispositius perquè canviï la configuració existent per la que nova.VII. Aquests envien la informació demanada cada interval de temps determinat per la configuracióVIII. Aquestes dades són desades en el servidor Kafka
<i>Extensions</i>	<p>En cas que l'administrador s'equivoqui en la configuració, l'aplicació retornarà un missatge d'error</p> <p>En cas que la connexió no pugui ser establerta entre l'aplicació i el servidor, retornarà un missatge d'error i no es realitzarà l'enviament.</p>

CAS D'ÚS #4: Aturar els resultats

<i>Actor</i>	Administrador del sistema
<i>Activador</i>	L'administrador aturar la monitorització existent
<i>Precondicions</i>	Hi ha una monitorització als dispositius
<i>Escenari principal</i>	<ol style="list-style-type: none">I. L'administrador indica la configuració que no vol cap configuració.II. Envia la informació als dispositius perquè parin el procés de monitorització actual.
<i>Extensions</i>	En cas que l'administrador s'equivoqui en la configuració, l'aplicació retornarà un missatge d'error En cas que la connexió no pugui ser establerta entre l'aplicació i el servidor, retornarà un missatge d'error i no es realitzarà l'enviament.

CAS D'ÚS #5: Visualitzar els resultats

<i>Actor</i>	Administrador del sistema
<i>Activador</i>	L'administrador desitja enviar una monitorització amb una configuració determinada a tots els dispositius registrats.
<i>Precondicions</i>	S'hagi enviat una configuració als monitors i aquests hagin enviat algun resultat
<i>Escenari principal</i>	<ol style="list-style-type: none">I. L'administrador consultarà directament sobre el servidor Kafka per veure quins són els resultats rebuts per un determinat tòpic.
<i>Extensions</i>	En cas de no haver resultats, retornarà buit En cas que la connexió no pugui ser establerta entre l'aplicació i el servidor, retornarà un missatge d'error i no es realitzarà l'enviament.

7. Requisits no funcionals

Els requisits no funcionals són aquells que jutgen el comportament d'un sistema de forma global i no de manera detallada com ho faria un requisit funcional. El projecte vol assolir els requisits no funcionals més importants, que són:

Fàcil d'utilitzar

<i>Descripció</i>	Ha de ser fàcilment usable
<i>Justificació</i>	Encara que la funcionalitat tingui una complexitat elevada, la interacció de l'usuari amb ella ha de ser simple ja que així reduïm els errors humans al màxim ja que l'usuari sabrà com s'ha de fer correctament.
<i>Criteri de satisfacció</i>	Assoliré el nostre requisit proveint a l'usuari d'exemples de com ha de realitzar el document de configuració, de manera que no es deixi cap element i la nomenclatura sigui la adequada i fent que l'usuari únicament necessiti un document JSON per executar tota la funcionalitat.

Fàcil d'aprendre

<i>Descripció</i>	Ha de ser fàcil aprendre a utilitzar l'aplicació
<i>Justificació</i>	La nostra aplicació ha de tenir un aprenentatge fàcil, ja que l'usuari no ha de perdre temps amb l'entrada de configuracions, sinó amb els resultats de sortida.
<i>Criteri de satisfacció</i>	L'aplicació pot ser utilitzada sense necessitat d'un manual, sense explicar res més que una plantilla de com ha de ser el format d'entrada de la configuració i la direcció del servidor.

Entendre la funcionalitat de l'aplicació

<i>Descripció</i>	L'usuari ha de saber quin és el resultat esperat donat un cert input.
<i>Justificació</i>	En l'aplicació és important saber quin és el resultat que vols per poder realitzar la configuració adequada.
<i>Criteri de satisfacció</i>	L'aplicació utilitza paraules relacionades amb la monitorització perquè l'usuari sàpiga sempre del que estem parlant.

Entendre la funcionalitat de l'aplicació

<i>Descripció</i>	L'usuari ha de saber quin és el resultat esperat donat un cert input.
<i>Justificació</i>	En la nostra aplicació és important saber quin és el resultat que vols per poder realitzar la configuració adequada.
<i>Criteri de satisfacció</i>	L'aplicació utilitza paraules relacionades amb la monitorització perquè l'usuari sàpiga quin tipus de valor és. A més oculto la implementació a l'usuari amb les crides RESTful al servidor

Velocitat i latència

<i>Descripció</i>	El sistema ha de ser ràpid i amb una latència baixa
<i>Justificació</i>	L'aplicació es basa en el temps, ja que rebem les dades que cert temps, si hi ha latència les dades no són en temps real, llavors no estem obtenint les dades que desitgem.
<i>Criteri de satisfacció</i>	Una vegada l'aplicació ha rebut la configuració, el resultat de les dades no hauria de trigar més d'un interval de temps de recollida de dades. Aquest temps el configura l'usuari perquè és el que necessita pel seu posterior estudi.

Precisió

<i>Descripció</i>	El sistema necessita donar uns resultats precisos
<i>Justificació</i>	La precisió en el resultat farà que els processos de després siguin més acurats i més verídics.
<i>Criteri de satisfacció</i>	Faré que tots els resultats tinguin tres decimals. A més de comprovar la unitat amb que són expressades les diverses mètriques.

Capacitat

<i>Descripció</i>	El sistema ha de ser d'alta capacitat, capaç de processar moltes dades.
<i>Justificació</i>	L'aplicació ha d'estar preparada per una alta capacitat perquè el número de dispositius que podem monitoritzar pot ser gran, per tant les dades obtingudes poden ser moltes en poc temps.
<i>Criteri de satisfacció</i>	Els servidors han d'estar muntats de manera que siguin concordants amb el número dades que processarà el sistema. A més de fer el sistema perquè hi hagin els mínims colls d'ampolla possible i en cas de que n'hi hagin minimitzar-los

Escalabilitat

<i>Descripció</i>	El nostre sistema ha de tenir la capacitat de poder créixer en cas de que augmentem el volum de dades.
<i>Justificació</i>	La nostra aplicació ha de ser capaç de poder tenir una alta escalabilitat en cas de que sigui necessari augmentar el volum de dades a processar.
<i>Criteri de satisfacció</i>	El codi està pensat perquè no importi el volum de dades. A més al tenir els diversos elements del sistema separats, permet que cada element sigui millorat si s'escau sense la necessitat que la resta d'elements se n'assabentin.

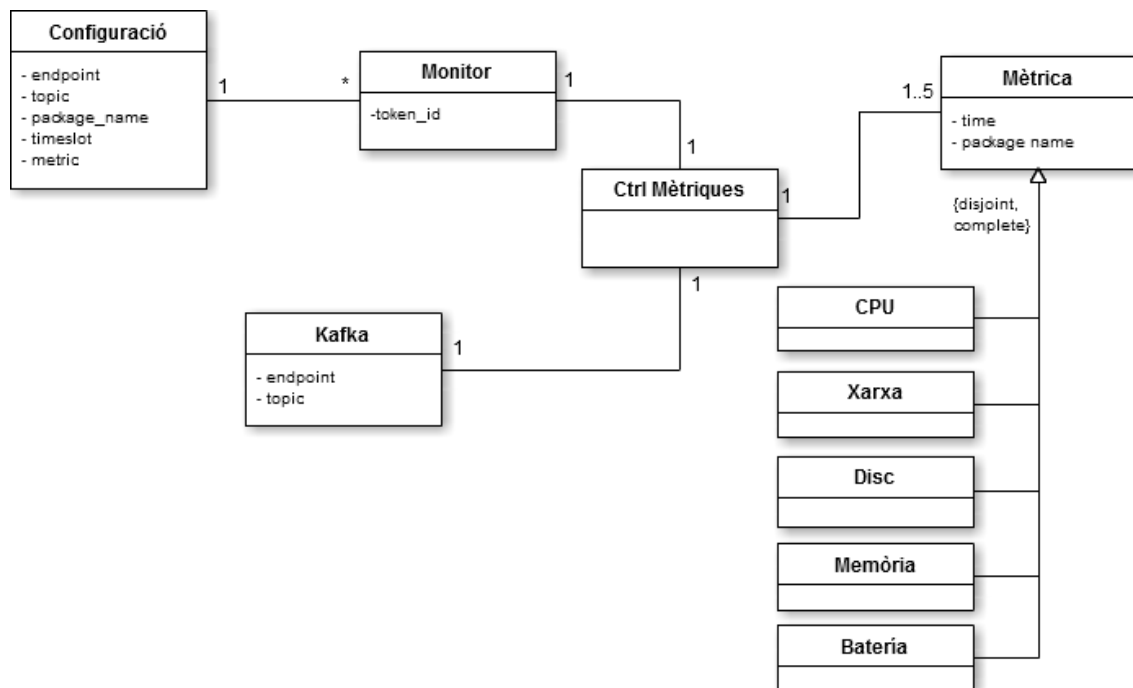
Fàcil manteniment

<i>Descripció</i>	El manteniment del codi ha de ser fàcil i ràpid
<i>Justificació</i>	Ja que segurament el manteniment no serà realitzat per la mateixa persona que ha realitzat el projecte, convé fer que el manteniment sigui una tasca fàcil i ràpida sense necessitat de la intervenció del desenvolupador del sistema.
<i>Criteri de satisfacció</i>	Hi haurà una reunió posterior a l'entrega del treball per poder explicar tot el codi exhaustivament amb la documentació del treball. A més, el codi és clar i pensat perquè sigui de fàcil enteniment.

8. Esquema conceptual

A partir dels diferents requisits funcionals i no funcionals he creat un esquema conceptual, que ens permetrà fer-nos una idea de com serà el nostre sistema sense entrar en massa detall en la implementació dels diferents elements.

L'esquema conceptual del projecte consisteix en que a partir d'una configuració, establerta pel usuari, s'envia a tots els monitors que estiguin al sistema. Aquest monitor s'encarregarà d'obtenir les mètriques demanades en la configuració. Per realitzar aquesta obtenció de dades, el monitor tindrà un controlador de mètriques, que serà l'element encarregat de cridar a les diferents mètriques per obtenir la dada demanada. Les mètriques que tindrem pel moment en el nostre sistema són cinc, cpu, xarxa, disc, memòria i bateria. Per últim, una vegada obtingudes les dades de les mètriques necessàries, aquest conjunt d'informació és enviat al servidor Kafka per ser emmagatzemat.



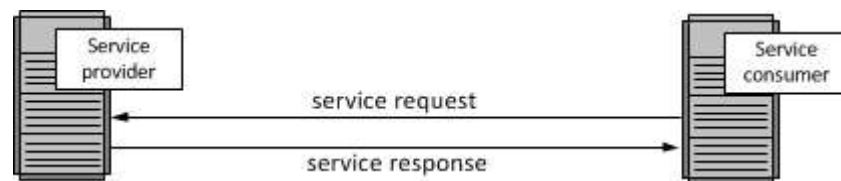
9. Disseny

a. Arquitectura lògica

L'arquitectura utilitzada per el projecte és una arquitectura orientada a serveis[13], entenent com a servei, una estructura ben definida, ben continguda i que és independent dels demés, fent que no depengui de l'estat dels altres serveis.

Aquest tipus d'arquitectura és una col·lecció de serveis que es comuniquen entre ells per realitzar una determinada funció, ja sigui una connexió entre dos elements o un simple enviament de dades.

La tecnologia més utilitzada per aquest tipus d'arquitectura és la del Web Services, on es realitza una petició i de manera asíncrona o síncrona es retorna una resposta. Aquests enviaments han de poder ser entesos per els elements de la connexió per a què aquesta sigui efectiva, per això, es recomana realitzar prèviament una unificació dels protocols, nomenclatures a utilitzar, tal i com hem realitzat en el nostre projecte.

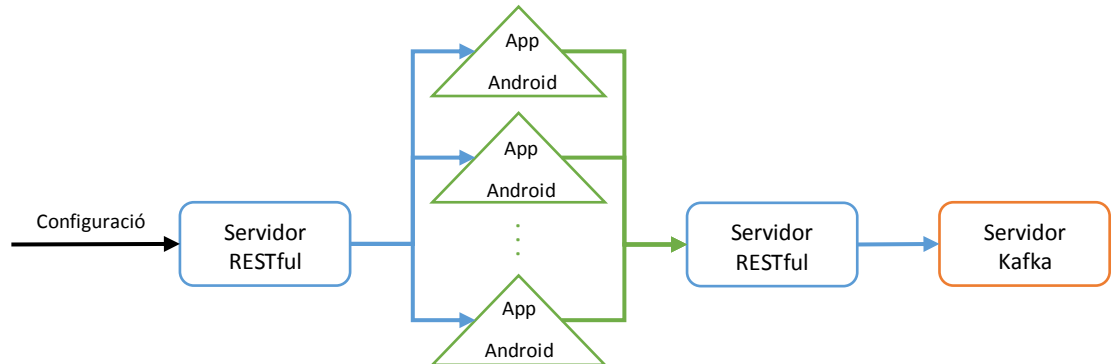


Aquest tipus d'arquitectura, és la més adient per el nostre projecte, com bé hem explicat, per dos grans raons. La primera és perquè ens permet un grau elevat d'independència entre els elements, fet que fa que puguem utilitzar, per exemple, diferents llenguatges de programació en el mateix projecte o que no sigui necessari conèixer les implementacions dels altres elements que no siguin el nostre.

La segona característica rellevant és la facilitat al canvi. És important tenir en compte que els requisits poden variar i el projecte ha de poder variar amb ell amb facilitat i procurant que s'hagi de tocar el mínim d'elements possibles del sistema.

b. Arquitectura física

L'arquitectura realitzada per aquest projecte consta de 4 elements principals, que són dos servidors RESTful, un servidor Apache Kafka i un aplicació mòbil Android, tal com podem observar a la figura de continuació.



Els servidors han de ser d'aquest tipus, RESTful, ja que al projecte SUPERSEDE ha estat especificats d'aquesta manera, igual que el servidor Kafka, per tant he hagut d'adaptar tot a aquestes limitacions. L'aplicació he decidit que sigui només per Android, ja que no tinc el temps ni els recursos suficients per poder desenvolupar una aplicació en IOS i Android.

Amb aquesta arquitectura, he hagut de fer diversos projectes per cada element del sistema, però en comptes de ser un inconvenient és un avantatge pel sistema. Com ja he esmentat anteriorment, al estar el projecte englobat en un projecte molt més gran, que els diferents elements estiguin clarament diferenciats, permet que alhora de realitzar algun tipus de canvi o substituir alguna part, la resta d'elements s'hagin de modificar mínimament, fent d'aquesta una avantatge molt important per el sistema. A més al ser elements diferents, permet que cada element sigui diferent, tant software com hardware, fent el sistema molt versàtil al canvi i la modificació.

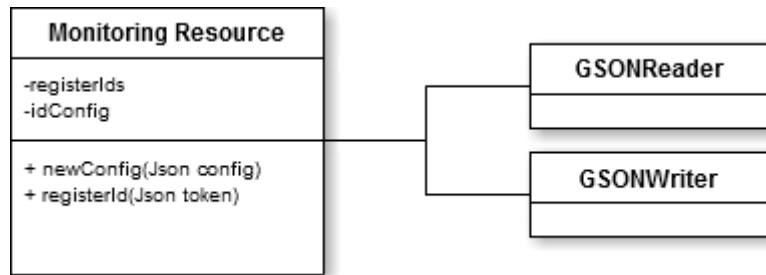
A continuació explicarem en més detall el disseny de cada element del sistema.

c. Servidor RESTful de Configuració

El servidor Apache Tomcat amb RESTful Service és l'encarregat d'enviar la configuració als dispositius mòbils registrats i de registrar els dispositius mòbils que volen rebre la configuració. A més, com podem observar en aquesta secció, els tres casos d'ús sobre les diferents funcionalitats relacionades amb els monitors, encès, modificació i apagament, han sigut ajuntades en una única funcionalitat, per a que a l'usuari, no al sistema, li sigui fàcil i simple la utilització del nostre sistema.

Diagrama de classes

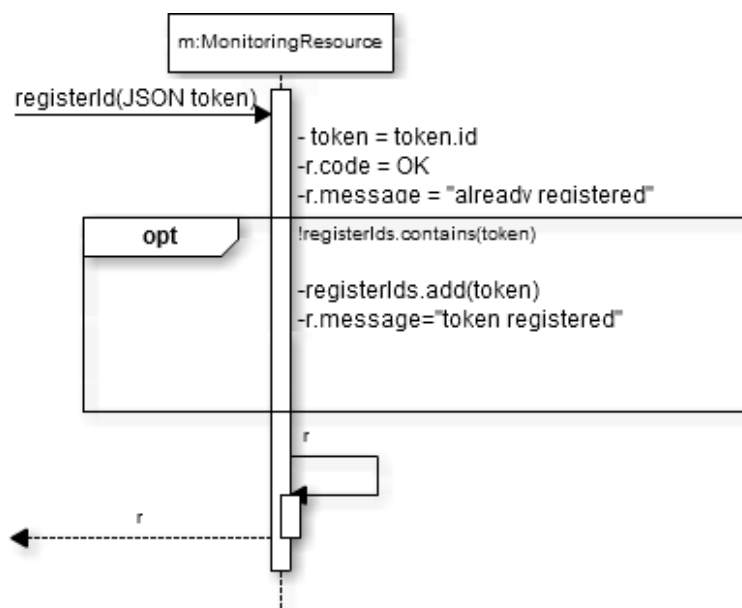
El diagrama de classes del servidor de configuració consta de 3 elements, el primer i principal, la classe Monitoring Resource és l'encarregada de l'enviament de configuració i el registre dels dispositius mòbils. Les altres dues classes, GSONReader i GSONWriter, són classes de suport que permeten a les crides del Servidor poder manegar documents JSON al seus paràmetres.



Diagrames de seqüència

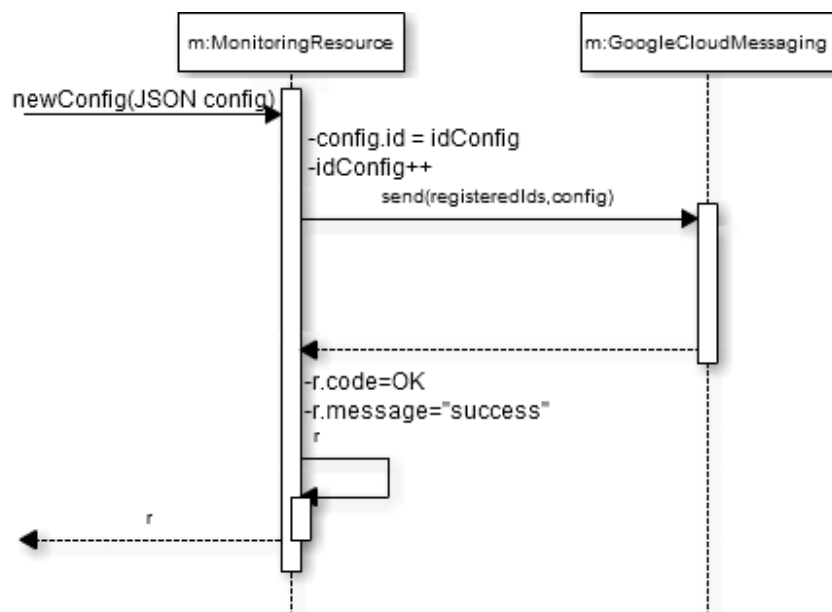
RegisteredId

És la operació que registra el identificador de les aplicacions en el nostre servidor, per posteriorment poder enviar la configuració a totes les aplicacions que estiguin registrades.



NewConfig

És la operació que envia la configuració desitjada a les aplicacions registrades.

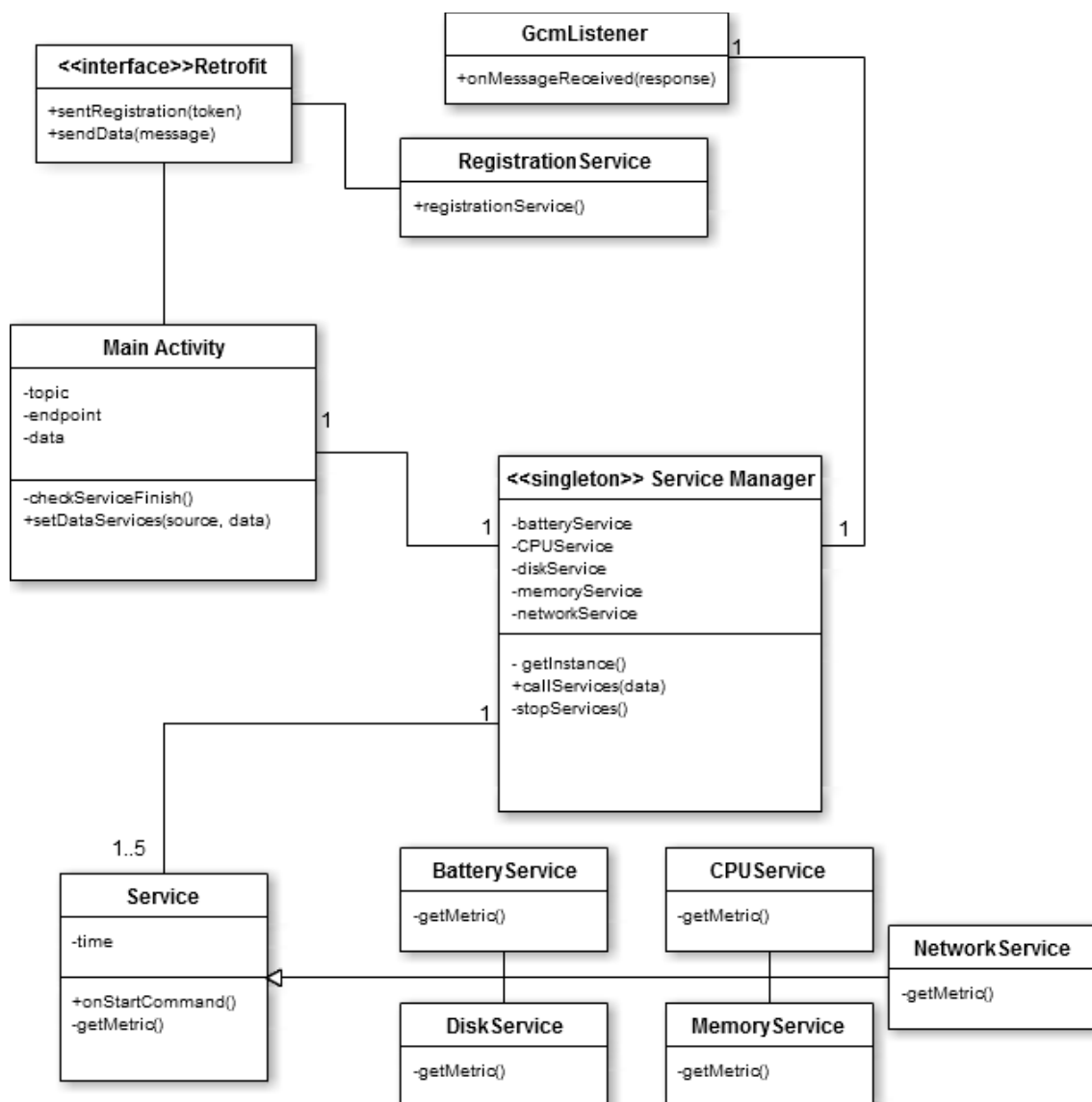


d. Aplicació de mòbil

L'aplicació mòbil és l'encarregada d'obtenir les dades que han demanat en la configuració per monitoritzada cada cert temps, també especificat en el document de configuració.

Diagrama de classes

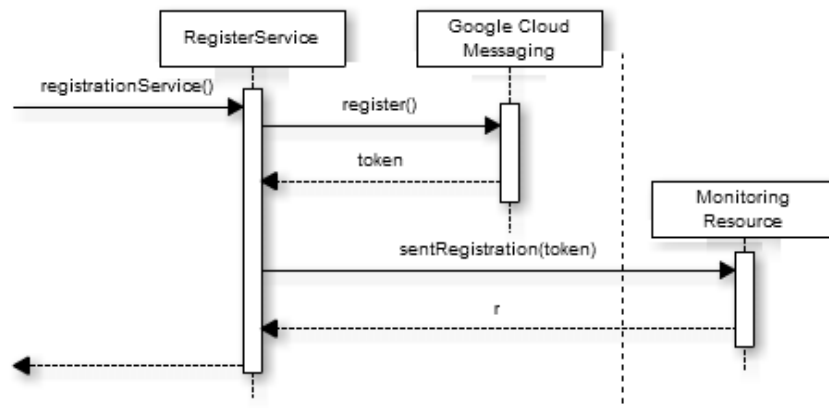
El diagrama de classes de l'aplicació mòbil consta d'onze classes, com podem observar a la figura. Tant la classe Registration Service, la MainActivity com la GcmListener s'encarreguen de la comunicació de l'aplicació amb l'exterior. Les dues primeres, amb l'ajuda de la llibreria Retrofit, són les que permeten realitzar la connexió cap a l'exterior, en canvi, el GcmListener, permet obtenir la configuració que és enviada a l'aplicació des del servidor. També tenim la classe Service Manager que és l'encarregada d'administrar els serveis, de cinc tipus diferents, a partir de la configuració que hagi arribat. Per últim, les classes de Serveis són les encarregades d'obtenir les dades que l'usuari necessita i enviar-les al MainActivity perquè aquest les envii al servidor d'emmagatzemament.



Diagrames de seqüència

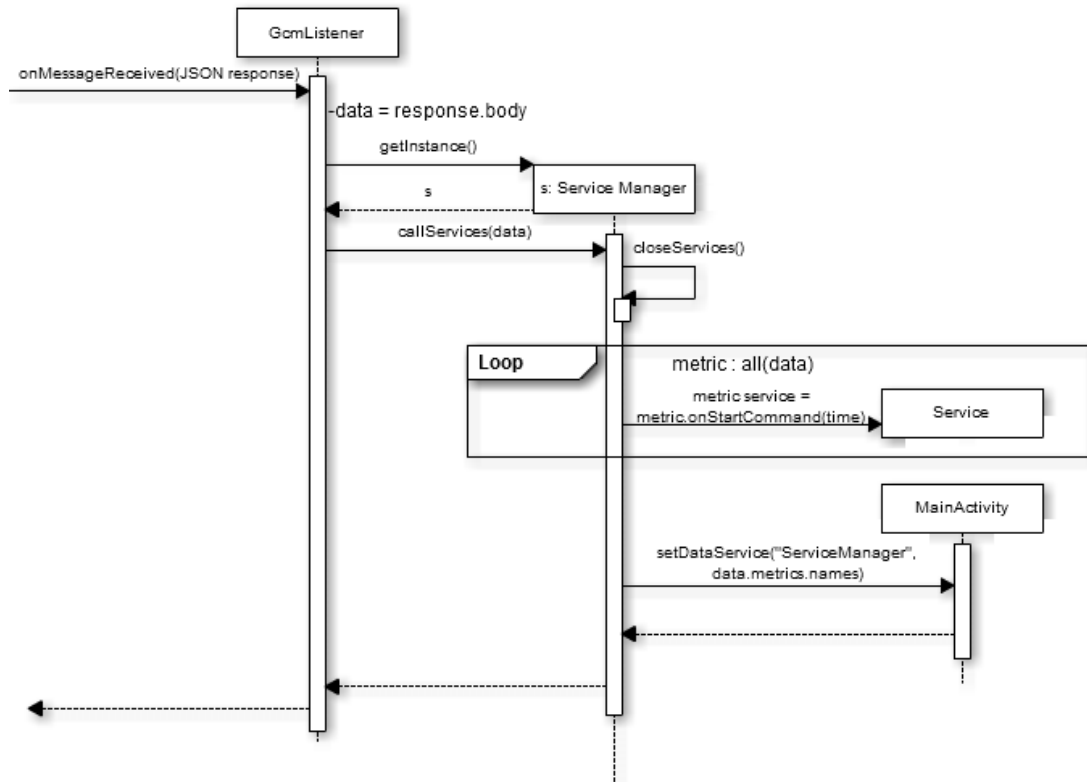
RegistrationService

És la operació que registre l'aplicació a Google i al nostre servidor.



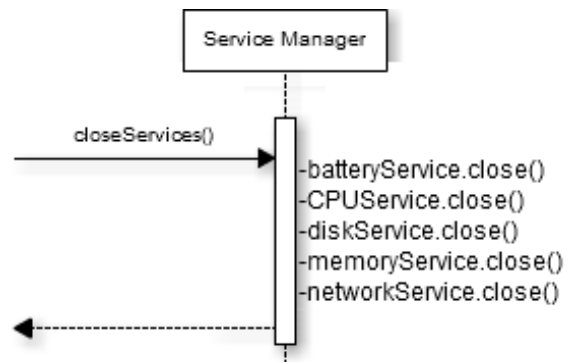
OnMessageReceived

És la operació que rep la configuració procedent del servidor i l'enviar al Service Manager. Aquest apaga els serveis que havien prèviament i encén aquells que han sigut demanats en la nova configuració. A més envia aquesta configuració al Main Activity ja també necessita la nova configuració



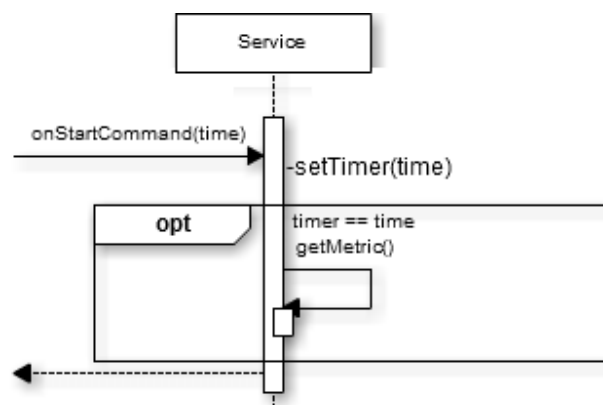
CloseServices

És la operació que para tots els serveis en execució.



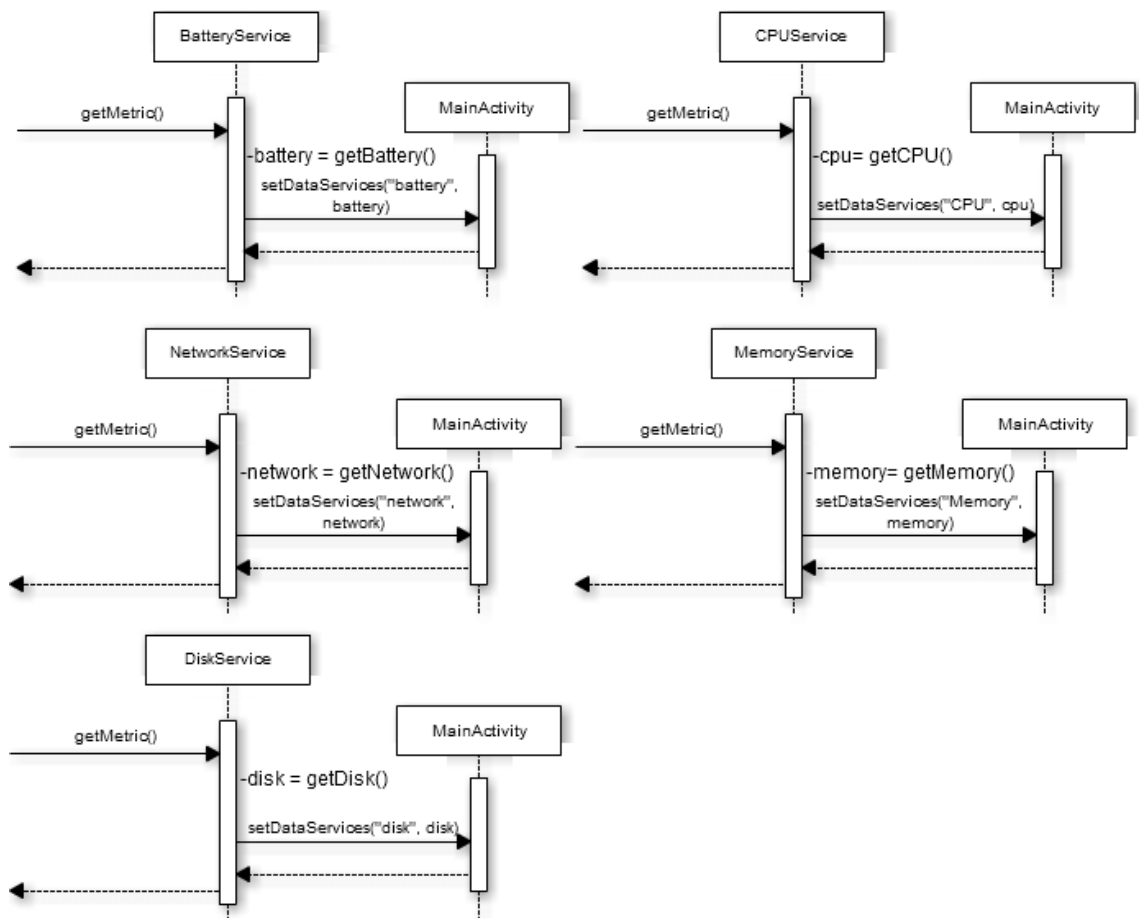
OnStartCommand

És la operació que inicia el servei perquè comenci a obtenir la mètrica necessària cada cert temps, el qual ha sigut especificat amb la configuració.



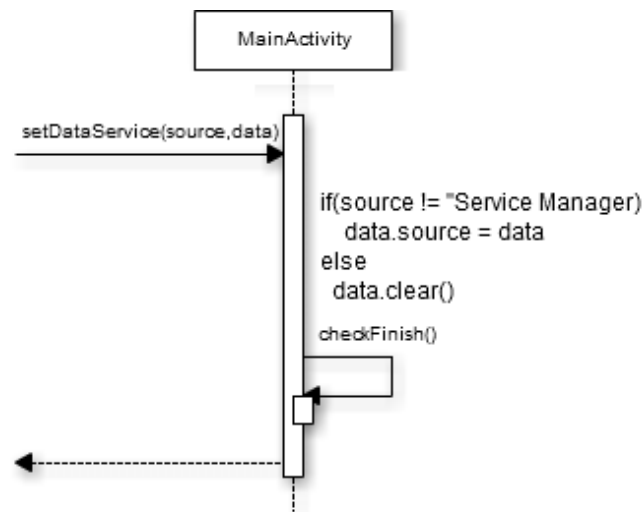
GetMetric

És la operació que obté la informació de cada servei i la envia al Main Activity perquè s'envii al servidor Kafka.



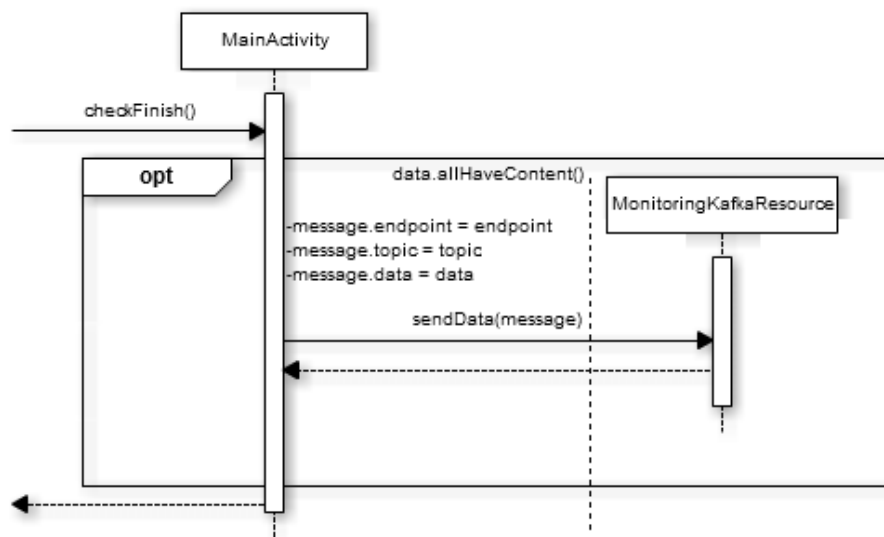
SetDataService

És la operació que guarda les dades de forma temporal de cada servei fins que tots els serveis actius hagin enviat les seves dades.



CheckFinish

És la operació que comprovar si tots els serveis actius han enviat les seves dades. Si és el cas, envia tota la informació al servidor perquè posteriorment l'envii al servidor Kafka.

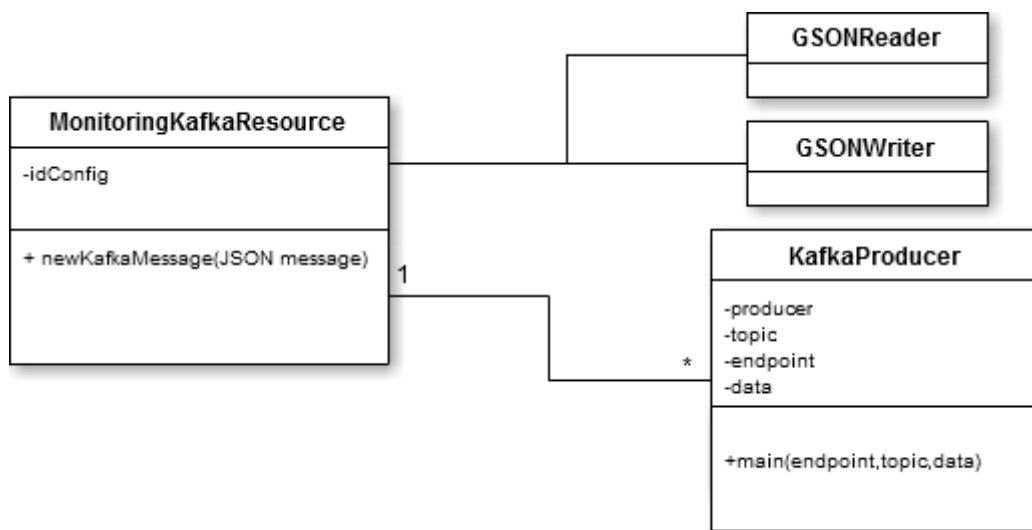


e. Servidor RESTful d'enviament a Kafka

Per últim, aquest element està compost per dos elements, ja que degut a un problema que va aparèixer durant la fase d'implementació, vaig haver de crear dos elements per poder realitzar l'acció d'enviar les dades obtingudes de les aplicacions al servidor Apache Kafka.

Diagrama de classes

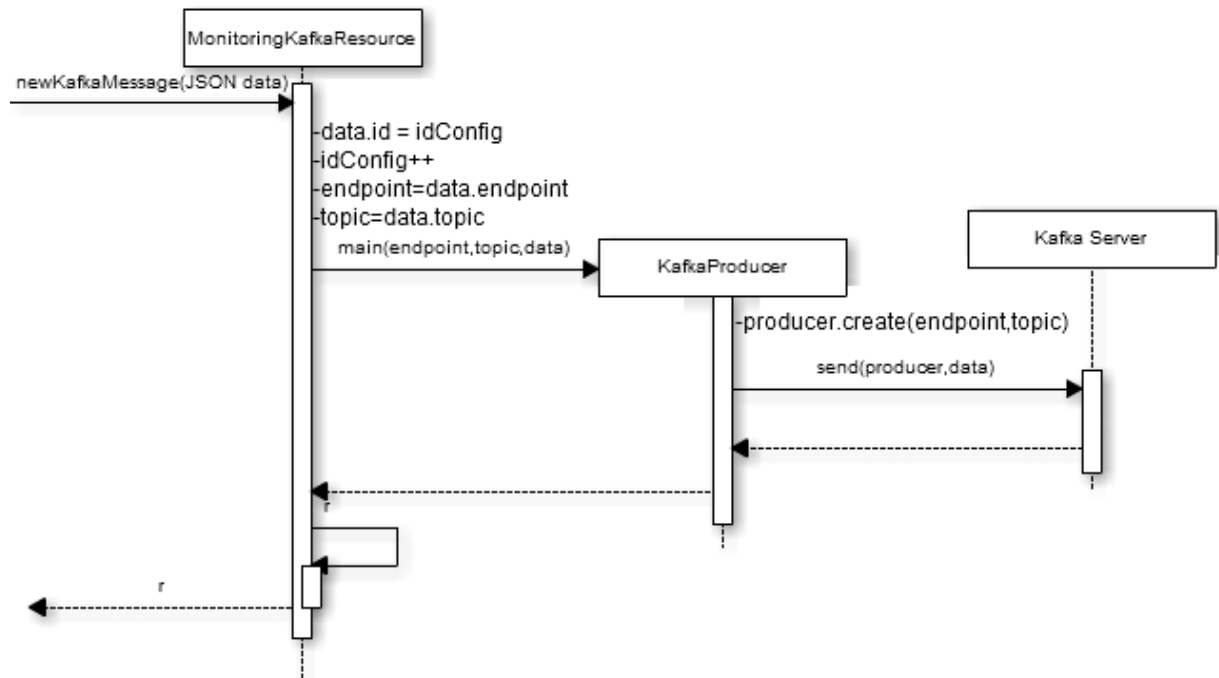
El diagrama del servidor d'enviament és bastant semblant al del primer servidor però en aquest diagrama tenim quatre classes. Com en el primer servidor, les classes GSONReader i GSONWriter són de suport per les crides del servidor RESTful. La classe MonitoringKafkaResource, és la principal i és la que rep la informació obtinguda a les aplicacions mòbils i executa la classe KafkaProducer, perquè aquesta enviï la informació rebuda al Servidor Kafka.



Diagrames de seqüència

newKafkaMessage

És la operació que envia la informació recollida per les aplicacions al servidor Kafka a l'adreça endpoint i tòpic especificats.



10. Implementació

a. Tecnologies utilitzades

En el projecte, les tecnologies utilitzades són molt diverses ja que el sistema està definit per diversos elements i cada element necessita una tecnologia diferent als demés. Les tecnologies utilitzades per el desenvolupament del nostre projecte són:

i. Google Cloud Messaging

Google Cloud Messaging és una tecnologia desenvolupada per Google que ens permet la comunicació entre el servidor i l'aplicació. Per poder utilitzar aquesta tecnologia, només es necessària un compte a Google.



Aquesta tecnologia, al ser molt utilitzada i bastant coneguda, té bastant suport pel desenvolupador, a més de tenir una petita tutoria per poder-lo instal·lar al teu sistema[10] i començar a veure com funciona per primer cop, cosa que ha fet disminuir la dificultat en l'aprenentatge.

En el projecte, l'he utilitzat en la comunicació entre el primer servidor RESTful, el que envia la configuració a les aplicacions, i els dispositius mòbils. Per poder implementar les funcionalitats en els servidors i en l'aplicació, hem tingut que registrar l'aplicació a Google i instal·lar la llibreria específica i un full de configuració per la aplicació. A més, a l'aplicació mòbil he hagut de donar d'alta aquest servei de missatgeria.

ii. Retrofit

Retrofit és una tecnologia molt similar a la anteriorment explicada, que ens permet comunicar-nos amb els servidors RESTful des de l'aplicació de mòbil.

Retrofit és una llibreria Open Source per Android que permet la comunicació entre una aplicació Android i un servidor RESTful Service.



En el projecte, l'utilitzo per realitzar les accions d'enviar el identificador de registre al servidor que envia la configuració de la monitorització, perquè sàpiga que el dispositiu està registrat, i per enviar les dades monitoritzades al servidor Kafka, perquè puguin ser emmagatzemades.

iii. Java

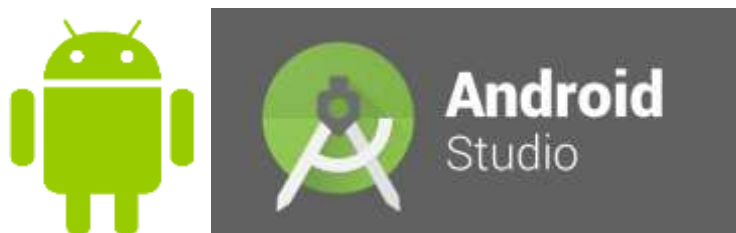
El llenguatge de programació Java ha sigut utilitzat per la implementació del Servidors. Per poder implementar de forma més còmode en Java hem utilitzat la plataforma Eclipse, que m'ha permès crear diversos tipus de projectes en funció de les característiques de cada element del sistema.



A més Eclipse, m'ha permès integrar els servidors de proves dins de la mateixa plataforma, fent que les proves siguin més senzilles de realitzar.

iv. Android

Per últim, el llenguatge per programar tota la part de l'aplicació mòbil és el llenguatge Android en la plataforma Android Studio, la qual em facilita el treball, a més de tenir molt suport, per les dificultats que em puc trobar durant la implementació del projecte.



b. Configuració de l'usuari

La configuració de l'usuari és la manera com el usuari interacciona amb el sistema, la manera amb la qual el sistema farà la monitorització que l'usuari desitja. Per això es important saber quin es el format d'aquest.

El primer que ha de tenir en compte l'usuari, és que el format de la configuració no ha sigut realitzat per nosaltres, sinó per altres persones del projecte que engloba el nostre projecte, perquè hi hagi una harmonia en el conjunt de documents necessaris en el sistema global. Segon, la introducció de les dades es realitza mitjançant una crida al servidor, a una direcció en concret, en aquesta crida hem d'adjuntar un document JSON amb un format com el de la figura següent.


```

{
  "endpoint": "Endpointprova5",
  "topic": "Topicprova5",
  "packageName": "Packageprova5",
  "time": "10000",
  "metrics": [
    {
      "metricName": "memory"
    },
    {
      "metricName": "cpu"
    },
    {
      "metricName": "network"
    }
  ]
}

```

Com podem veure a la figura, per poder realitzar correctament el JSON de configuració necessitem tenir una sèrie d'elements:

- Un node endpoint, el qual tindrà la direcció del servidor Kafka on enviarem les dades de la monitorització
- Un node topic, que ens indicarà en quin node dins del servidor Kafka voldrem deixar les dades de la monitorització
- Un node packageName, on tindrem el nom del procés que volem monitoritza dels nostres dispositius mòbils.
- Un node time, per indicar la freqüència en que obtenim la informació dels dispositius en milisegons
- Un node metrics, on tindrem les mètriques que necessitem per la nostra monitorització. Actualment només tenim 5 possibilitats que són:
 - Memory, per obtenir la memòria del dispositiu mòbil
 - CPU, per obtenir la CPU del dispositiu mòbil
 - Network, per obtenir les dades de xarxa utilitzades
 - Battery, per obtenir el nivell de bateria del dispositiu mòbil
 - Disk, per obtenir la capacitat del disc

c. Funcionalitats desenvolupades

Tot el sistema realitza una única funcionalitat general, la capacitat de monitoritzar un dispositiu mòbil, però per fer més entenedor el treball realitzat, la divisió de funcionalitats seran les diverses implementacions dels elements del sistema.

En el sistema tenim 3 elements lògics principals, però com ja veurem més endavant, alhora de la implementació he tingut que separar un element en dos parts, fent que hagi un conjunt de 4 elements.

He decidit fer aquesta separació, ja que cada element és un projecte diferent, per tant, ja hi ha una separació física . A més cada element del projecte té unes determinades característiques diferents als altres i m'agradaria que quedessin ben diferenciades.

Aquestes tres funcionalitats o elements del sistema diferents són:

1. Servidor RESTful d'enviament de la configuració

Aquest element va ser el primer en ser implementat, ja que havia de realitzar una mica de recerca al respecte sobre les tecnologies necessàries i com realitzaria el servidor, al no tenir experiència prèvia de projectes similars en aquest llenguatge ni en la plataforma on l'anàvem a programar. La part positiva és que el codirector del treball, en Marc Oriol Hilari, tenia experiència en aquest tema i va ajudar a que la recerca fos més lleugera i més fàcil.

Abans de començar amb el projecte, vaig començar amb uns petits projectes tests. Amb l'ajuda de diverses webs vaig poder realitzar-los, fet que em va ajudar entendre com funcionava la realització d'un servidor RESTful en Eclipse.

A partir d'aquí, vaig començar a implementar el projecte amb les restriccions imposades. La primera de totes, el servidor havia de ser un RESTful Service, per tant havia de ser un tipus de projecte molt específic i el servidor vinculat havia de ser un determinat, un Apache Tomcat de la versió més actualitzada fins aquell moment, en aquest cas la versió 8.

Una vegada vaig tenir l'entorn muntat, vaig començar a programar les diferents funcionalitats amb l'ajuda dels projectes test que havíem realitzat prèviament. Com ja he esmentat en seccions anterior, en aquest element del sistema, tenim dos funcionalitats la de registrar el dispositiu mòbil i la d'enviar la configuració als dispositius mòbils, aquesta última va ser la elecció escollida per començar, ja que així podria anar provant la comunicació entre el servidor i l'aplicació, ja que la comunicació a l'inrevés no la podria provar amb l'aplicació fins que l'implementés i només la podria provar amb el Postman, programa que realitza crides REST sobre la direcció que l'indiques.

Per realitzar aquesta primera funcionalitat d'enviament de configuracions, el primer que vaig fer va ser instal·lar totes les llibreries necessàries i vam començar a fer probes des del Postman per veure que les crides realitzades tenien el resultat esperat. A partir d'aquí vam començar a programar la part de la comunicació amb les aplicacions mòbils i em vaig adonar que necessitava alguna aplicació, encara que fos sense cap funcionalitat, que rebés la petició. Per tant,

vaig implementar aquesta petita part de l'altre element, que explicaré en el següent punt i vaig continuar amb el primer servidor.

Al tornar al servidor, vaig començar a provar de nou, però em vaig trobar amb un petit imprevist i es que el JSON que necessitava enviar, no era correcte. Vaig fer una mica de recerca i amb l'ajuda del codirector, em va recomanar el canvi de llibreries de JSON. Al canviar de llibreries, vaig haver de realitzar unes classes noves per a la llibreria, que consistien en classes de lectura i d'escriptura de JSON. Després de realitzar aquest canvi, va funcionar la nostra crida a les aplicacions mòbils i vaig comprovar que totes les dades enviades eren les adequades i eren les que necessitava.

La segona funcionalitat, el registre dels dispositius mòbils al servidor, va ser implementada al acabar l'anterior. Aquesta funcionalitat no va tenir cap complicació, ja que l'única cosa que havia de fer era rebre l'identificador del dispositiu i emmagatzemar-lo en cas que de no tenir-lo. Encara que havia acabat la implementació, no havia acabat, em faltava poder provar-la de forma completa. Havia estat provada mitjançant la utilització del Postman però havia d'assegurar-me que funcionava exactament igual amb una crida des d'un dispositiu mòbil i aquesta part no la vam poder realitzar fins que l'aplicació mòbil no va avançar una mica i vam implementar la crida a registre. Amb la crida enllestida, vaig provar i corregir els petits detalls que faltaven, per així deixar enllestit el servidor de configuració.,

II. Aplicació Android del dispositiu mòbil

Aquest segon element, com he dit en l'apartat anterior, va haver de ser implementat abans de l'esperat ja que necessitava les funcions de comunicació del servidor.

Prèviament a començar amb el desenvolupament, vaig realitzar unes aplicacions test amb el codi proporcionat per Google, el qual em donava un programa de demostració per provar la seva llibreria de comunicació, Google Cloud Messaging, la qual he explicat anteriorment. Al veure que l'aplicació de demostració aplicant-hi els canvis necessaris podria complir amb els requeriments, vaig continuar amb la següent part.

Vaig procedir a implementar una primera part de l'aplicació que em faria poder avançar amb la part del primer servidor, la comunicació exterior amb l'aplicació, però no a l'inrevés.

Després de realitzar la primera fase, vaig començar la segona fase. Vaig començar investigant la millor manera per realitzar la monitorització de l'aplicació. Al acabar, vam decidir que necessitava

quelcom que no requerís de l'aplicació oberta a l'usuari per funcionar. Això em va dur a la implementació de serveis.

Al utilitzar serveis, la implementació que tenien pensada havia de canviar, ja que cada servei és independent de l'altre. Els serveis són fluxos que s'executen de forma autònoma i s'executen independentment si la aplicació està activa visualment a l'usuari o no. Per tant, per poder administrar correctament aquests serveis, vaig requerir d'una classe addicional que la vaig anomenar Controlador de Serveis. De manera que els serveis estiguessin el més desacoblats de l'aplicació possible, així els podria modificar de forma fàcil i ràpida, ja que segurament en un futur es vulguin implementar nous serveis i modificar-ne existents.

Una vegada realitzat vaig començar a pensar com fer per comunicar-se amb els servidors RESTful. Recercant llibreries d'Android, vaig trobar Retrofit, una llibreria que em permetia fer crides RESTful als servidors des de l'aplicació mòbil. A partir d'aquí vaig realitzar la implementació d'aquesta llibreria, la qual només requeria l'afegit d'una petita interfície on podem trobar les crides al servidor.

Una vegada implementat, vaig deixar de banda un moment aquest element, per posar-me amb la última part del primer servidor i deixar enllestida la comunicació.

Una vegada acabat la implementació del primer servidor, vaig procedir a continuar amb l'aplicació mòbil. L'únic que mancava per acabar en el nostre element era la recollida de les dades obtingudes dels serveis i el seu enviament. Per dur-ho a terme, em va ajudar la manera com havia implementat els serveis, ja que els serveis només són accessibles per el controlador de serveis. Encara que es el controlador l'encarregat del servei, aquest només pot crear-los i eliminar-los, per tant la comunicació amb els serveis es fa amb una comunicació interna que té Android. A més relacionat amb el tema, em vaig trobar que els serveis eren independents un dels altres per tant eren asíncrons, llavors havia de pensar una manera de poder enviar les dades de tots els serveis a la vegada.

La solució que vaig trobar consisteix en esperar a que tots els serveis en execució enviïn les seves dades d'una iteració per més tard poder enviar-les de tots els serveis en el mateix bloc. Això va fer que modifiqués la manera com l'usuari envia la configuració del temps, com expliquem en l'apartat del document de configuració. Una vegada tenim totes les dades del cicle, generem un document de tipus JSON per poder-ho enviar cap a l'altre servidor.

En aquest punt, vaig deixar la implementació fins que no tingués el servidor per poder provar correctament, havia deixat tot preparat, connexió amb Retrofit i JSON a enviar, però a la espera de fer la última connexió.

Per últim i terminada la part que necessitava de l'altre servidor, vaig provar que la comunicació i les dades eren correctes, per deixar-ho ja enllestit i acabat.

III. *Servidor RESTful d'emmagatzemament a Kafka*

Aquest element és l'últim implementat i aparentment era el que pensava que menys treball em portaria, ja que la funcionalitat que havia de realitzar era la d'enviar les dades rebudes dels dispositius mòbils al servidor Kafka, aparentment una funcionalitat senzilla.

Una vegada realitzat la implementació del primer servidor, aquesta havia de ser bastant similar. Com en els elements anteriors, vaig començar primerament amb la investigació sobre com comunicar el nostre servidor amb el servidor Kafka, al qual havia d'enviar les dades. Una vegada realitzada la recerca, vaig alguns programes de prova per veure el funcionament. Al realitzar aquests, el que ens vaig adonar va ser que cap em funcionava per el que volia fer, ja que les llibreries que necessitava utilitzar no em funcionaven correctament en la solució. Amb aquest problema al cap, vaig estar pensant com podia realitzar la solució d'una altra manera. Per la meua sorpresa, em vaig adonar que si implementava un projecte separat, aquest projecte permetia enviar les dades a Kafka però si l'integrava dins del servidor no funcionava. Tenint aquestes premisses, el que vaig fer, va ser que el servidor només funcionés d'intermediari i que fos una aplicació la que realitza l'enviament de dades, d'aquesta manera podria solucionar el problema.

Amb la nova solució en ment, vaig procedir a implementar-la en el servidor. Primerament vaig muntar el servidor i després vaig crear un projecte nou amb la aplicació que envia les dades a Kafka. Aquesta nova visió em permetia solucionar el nostre problema, a més de donar-nos una independència entre el servidor i l'enviament de manera que si volen modificar, en un futur, perquè funcioni d'una altra manera la funcionalitat del servidor seguiria funcionant exactament igual. A més, al haver d'executar l'aplicació permet la millor paral·lelització, ja que s'executa una aplicació per enviament.

Amb aquesta última part ja tenia els tres elements, quatre si tenim en compte l'aplicació per l'enviament de dades a Kafka, ara només tocava la fase de proves amb tot el conjunt del sistema. Encara que ja havia realitzat proves parcials per cada element.

11. Validació

a. Entorn de proves

Com a tot projecte, totes les funcionalitats han de ser provades, per veure que tot funciona correctament i de la forma desitjada. En el meu cas, les proves s'han fet en un entorn local, ja que la necessitat de servidors ha fet que sigui dificultós el muntatge d'un sistema de proves per només provar una persona, el desenvolupador del projecte. Per tant, he hagut de muntar-me un petit sistema en un sol ordinador per simular tot el flux del sistema.

El fet que el sistema sigui en un ordinador i no en uns servidors, no ha fet que variés el projecte o que els resultats esperats siguin diferents, al contrari el nivell de qualitat en les proves segueix sent el mateix que si les fes en uns servidors dedicats. L'únic inconvenient és que les proves són una mica més feixugues, ja que el principal problema és que l'aplicació de mòbil necessita poder accedir als servidors i com que aquests es troben dins del ordinador, he de procurar donar una sèrie de permisos de xarxa al router domèstic perquè pugui poder fer aquestes crides al servidor que es troba dins l'ordinador.

Per últim, per ajudar-me en el procés de proves, he utilitzat el programa Postman[12], ja esmentat anteriorment, que és una aplicació que et permet llençar crides RESTful sobre la direcció que tu desitgis amb els paràmetres que tu necessitis. Amb aquesta eina em permet simular qualsevol tipus d'input que necessiti per la nostra aplicació.

b. Pla de proves

Les proves realitzades s'han realitzat de manera manual i per cada element del sistema per poder-nos assegurar que cada element funciona perfectament sense els demés elements. A més de provés a posteriori del conjunt de sistema amb tots els elements junts.

En aquestes proves el que he intentat realitzar és la realització de tots els casos necessaris per poder abastar la màxima diversitat i així poder controlar el màxim rang possible.

En el cas dels elements que no són el primer servidor, que és el que rep la informació que dona l'usuari, per la realització de les proves individuals de cada element, he simulat la informació que l'aplicació hauria de transmetre a aquesta part del sistema, comprovant que totes les dades que envia es transfereixin correctament i no hi hagi cap tipus de desincronització.

c. Informe de proves

Per aquestes proves, he realitzat un petit informe propi per a cada cas d'ús i cada diferent casuística d'aquest, com el que tenim a continuació, ja que totes les proves eren de forma manual i no automatitzada. Addicionalment vam crear va ser una sèrie de tiquets a la wiki, una funcionalitat que em proporciona el programa de versions, Bitbucket, per deixar constància de l'error que havíem trobat, per més tard poder arreglar-lo.

A continuació podrem trobar alguns informes propis realitzats per a les proves.

Element del sistema – Cas d'ús	Dades d'entrada	Resultat	Tester
<i>Servidor de configuració – Registrar Dispositiu</i>	Registrar un dispositiu per primera vegada	OK	RSO
<i>Servidor de configuració – Registrar Dispositiu</i>	Registrar un dispositiu que ja està registrat	OK	RSO
<i>Servidor de configuració – Configurar monitors – Enviar una configuració</i>	Input a Postman: <pre>{ "endpoint": "Endpointprova5", "topic": "Topicprova5", "packageName": "Packageprova5", "time": "10000", "metrics": [{ "metricName": "memory" }, { "metricName": "cpu" }, { "metricName": "network" }] }</pre>	OK	RSO
<i>Dispositiu mòbil – Configurar monitors – Parar els monitors</i>	Input a Postman: <pre>{ "endpoint": "Endpointprova5", "topic": "Topicprova5", "packageName": "Packageprova5", "time": "10000", "metrics": [] }</pre>	OK	RSO
<i>Dispositiu mòbil – Configurar monitors – Modificar el temps</i>	Input a Postman: <pre>{ "endpoint": "Endpointprova5", "topic": "Topicprova5", "packageName": "Packageprova5", "time": "30000", "metrics": [{ "metricName": "memory" }, { "metricName": "cpu" }, { "metricName": "network" }] }</pre>	OK	RSO

<p><i>Servidor Kafka – Visualitzar resultats</i></p>	<p>Input a Postman:</p> <pre>{ "nameValuePairs": { "endpoint": "localhost:9092", "topic": "test", "timeStamp": "Dec 10, 2016 4:00:02", "numDataItems": 3.0, "metrics": { "values": [{ "nameValuePairs": { "name": "memory", "available": 30.622562408447266, "package Total PSS": 0.0, "package Total Shared Dirty": 0.0, "package Total Private Dirty": 0.0 } }, { "nameValuePairs": { "name": "network", "transmitted": 27.0, "received": 511.0, "package transmitted": 0.0, "package received": 0.0 } }, { "nameValuePairs": { "name": "cpu", "available": 18.604650497436523 } }] } } }</pre>	OK	RSO
<p><i>Tots – Configurar monitors – Enviar una configuració</i></p>	<p>Input a Postman:</p> <pre>{ "endpoint": "Endpointprova5", "topic": "Topicprova5", "packageName": "Packageprova5", "time": "10000", "metrics": [{ "metricName": "memory" }, { "metricName": "cpu" }, { "metricName": "network" }, { "metricName": "battery" }, { "metricName": "disk" }] }</pre>	OK	RSO

12. Planificació

a. Duració estimada prèvia del projecte

El projecte té una durada estimada de 4 mesos. L'inici del projecte serà l'1 de setembre i la finalització d'aquest serà el 31 de desembre aproximadament.

b. Consideracions prèvies

Al utilitzar una metodologia Agile, totes les planificacions poden canviar ja que potser necessito quelcom al projecte que no estava previst. Una altre consideració a tenir molt en compte és el fet que el projecte està sotmès a un calendari fix que dona la Facultat de Informàtica de Barcelona i no pots estendre'l.

c. Fases del projecte

En aquestes fases no hi haurà fase de documentació prèvia ja que al ser un projecte previ ja la té, per tant començarem directament amb la fase de programació i documentació de l'assignatura de GEP.

i. Fase de gestió del projecte

Aquesta fase és la fase que realitzaré durant l'assignatura de GEP, on hauré d'especificar l'abast del nostre projecte, la planificació temporal, la econòmica, etc. Com no tenim temps a fer-ho abans de la fase de desenvolupament, aniré fent-ho a la vegada.

ii. Fase de desenvolupament

Aquesta fase es basa en la implementació dels monitors sobre l'aplicació, dins d'aquesta fase també englobem el fet de l'aprenentatge autònom. Dividiré aquesta secció en els diferents elements que he d'implementar i no en les iteracions que realitzaré, perquè dins d'una mateixa iteració puc fer dues tasques de llocs diferents.

Servidor per les configuracions

Hauré de programar un servidor RESTful perquè pugui rebre les peticions post amb la configuració i pugui enviar aquesta configuració a totes les aplicacions connectades.

Aplicació per realitzar la monitorització

Hauré d'implementar l'aplicació perquè pugui rebre la configuració adient del servidor i programar els les diferents mètriques a monitoritzar

Servidor per enviar les dades extretes

En aquesta última fase del desenvolupament, consisteix en la implementació d'un servidor Apache Kafka que rebrà la informació monitoritzada de les diferents aplicacions mòbils.

iii. Fase de documentació

Aquesta darrera fase serà la conclusió del nostre projecte, ja que documentarem tot el treball realitzat utilitzant de guia el que he estat realitzant durant l'assignatura de GEP. Aquest treball previ realitzat em permetrà un guany en temps ja que em donarà una idea de com fer l'estructura del nostre treball.

Aquestes fases es realitzaran en un ordre determinat per poder realitzar el treball de forma més eficient. Per tant començaré, primerament, amb la creació del Servidor RESTful per poder fer les crides. Posteriorment, començaré amb la comunicació entre servidor i aplicació de mòbil i el nucli de l'aplicació. Finalment acabaré amb el Servidor Kafka per poder bolcar la informació extreta. Per finalitzar el projecte, realitzaré tota la documentació necessària pel projecte.

d. Temps estimats

Temps estimats per fase

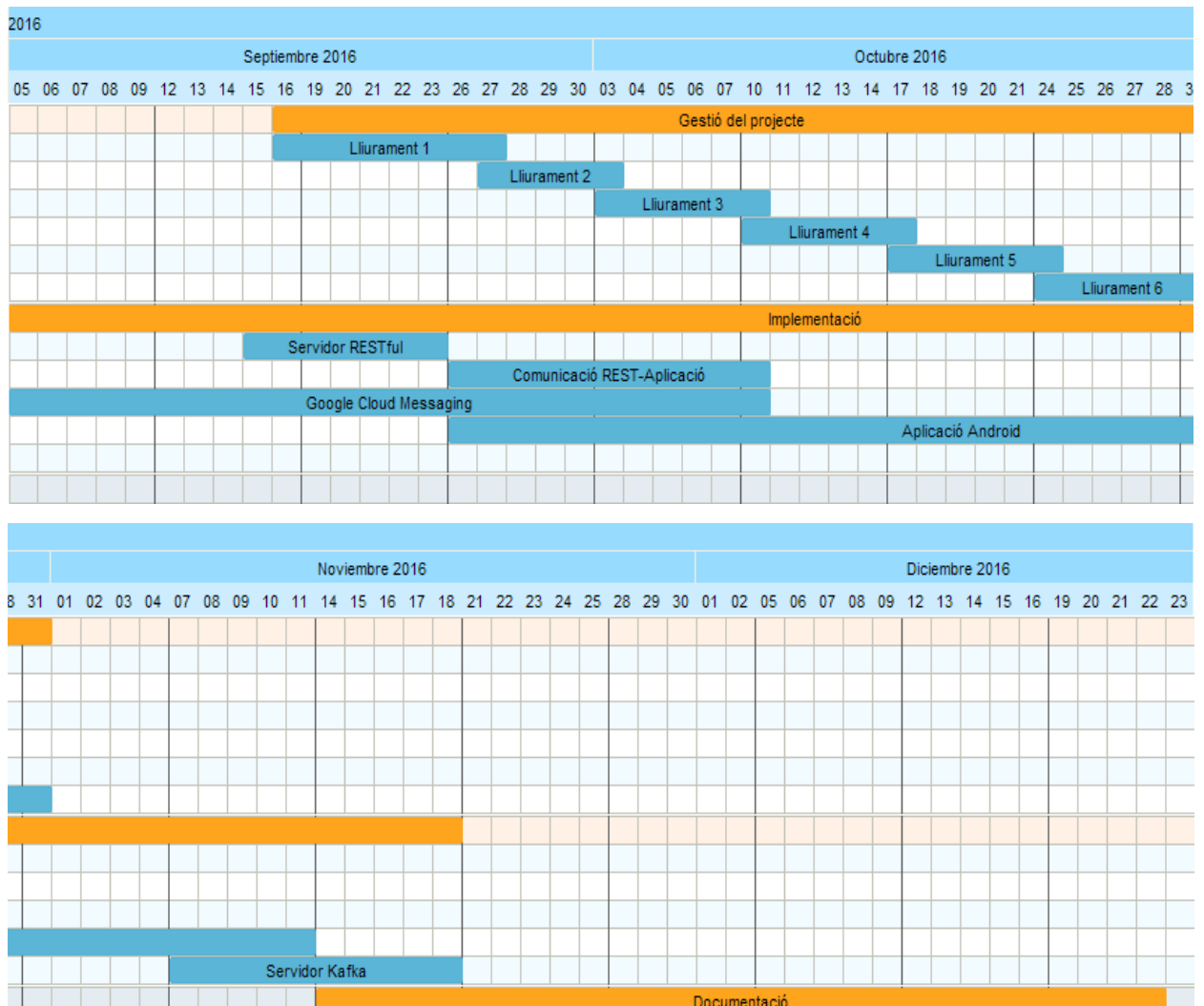
Tasca	Temps estimat (hores)
Fase de gestió del projecte	75
Fase de desenvolupament	400
Servidor RESTful	100
Aplicació Android	200
Servidor Kafka	100
Fase de documentació	30
Total temps invertit	505

Temps estimats per fase i tipus de membre del projecte

Tasca	Temps estimat (hores)
Fase de gestió del projecte	75
Cap del projecte	75
Fase de desenvolupament	400
Servidor RESTful	100
Cap del projecte	5
Desenvolupador	70
Responsable de les proves	25
Aplicació Android	200
Cap del projecte	5
Desenvolupador	150
Responsable de les proves	45
Servidor Kafka	100
Cap del projecte	5
Desenvolupador	80
Responsable de les proves	15
Fase de documentació	30
Cap de projecte	30
Total temps invertit	505

e. Diagrama de Gant

El diagrama de Gant conté les tasques esmentades anteriorment. Ho dividim en dues parts per a que sigui més fàcil la visualització.



f. Pla d'acció

Com a tots els projectes, he de tenir en compte la possibilitat de trobar-me amb obstacles durant la realització del projecte. Per això, la possibilitat de problemes s'ha de tenir en compte durant la planificació del projecte.

El projecte en principi t'he previst acabar abans de les vacances de nadal, 22 de desembre. Aquesta data està posada de manera que si es necessari més temps per la realització de la documentació del projecte tingui el temps suficient per poder reaccionar abans de la data definitiva d'entrega.

Per altra banda, en cas de tenir un problema amb la implementació, ja sigui per un error o per la dificultat en l'aprenentatge, les setmanes de desembre estan programades com a setmanes de reserva en cas que necessiti més temps per aquesta part del projecte.

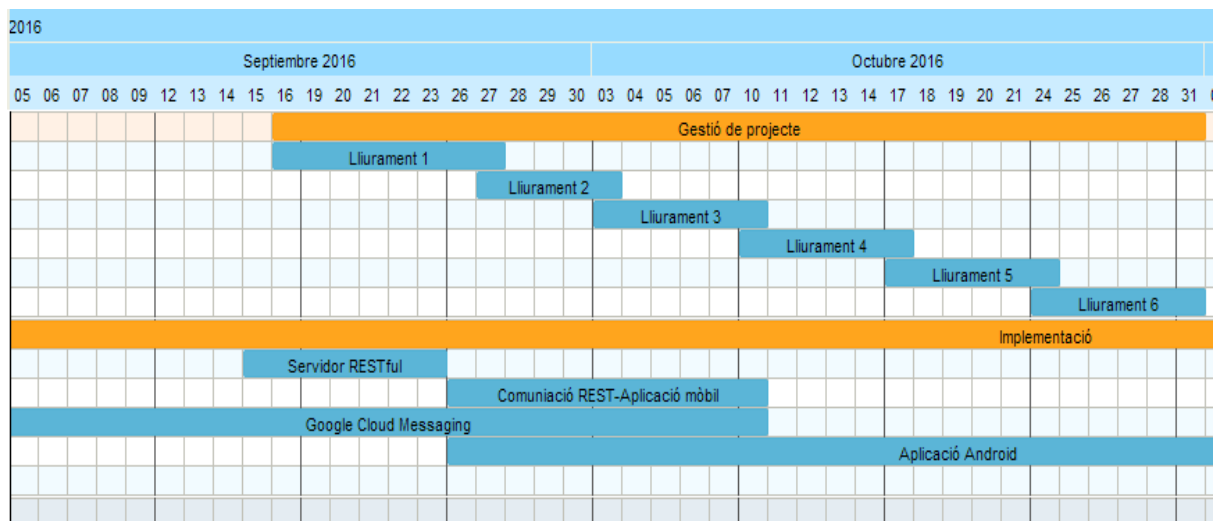
g. Desviacions ocorregudes

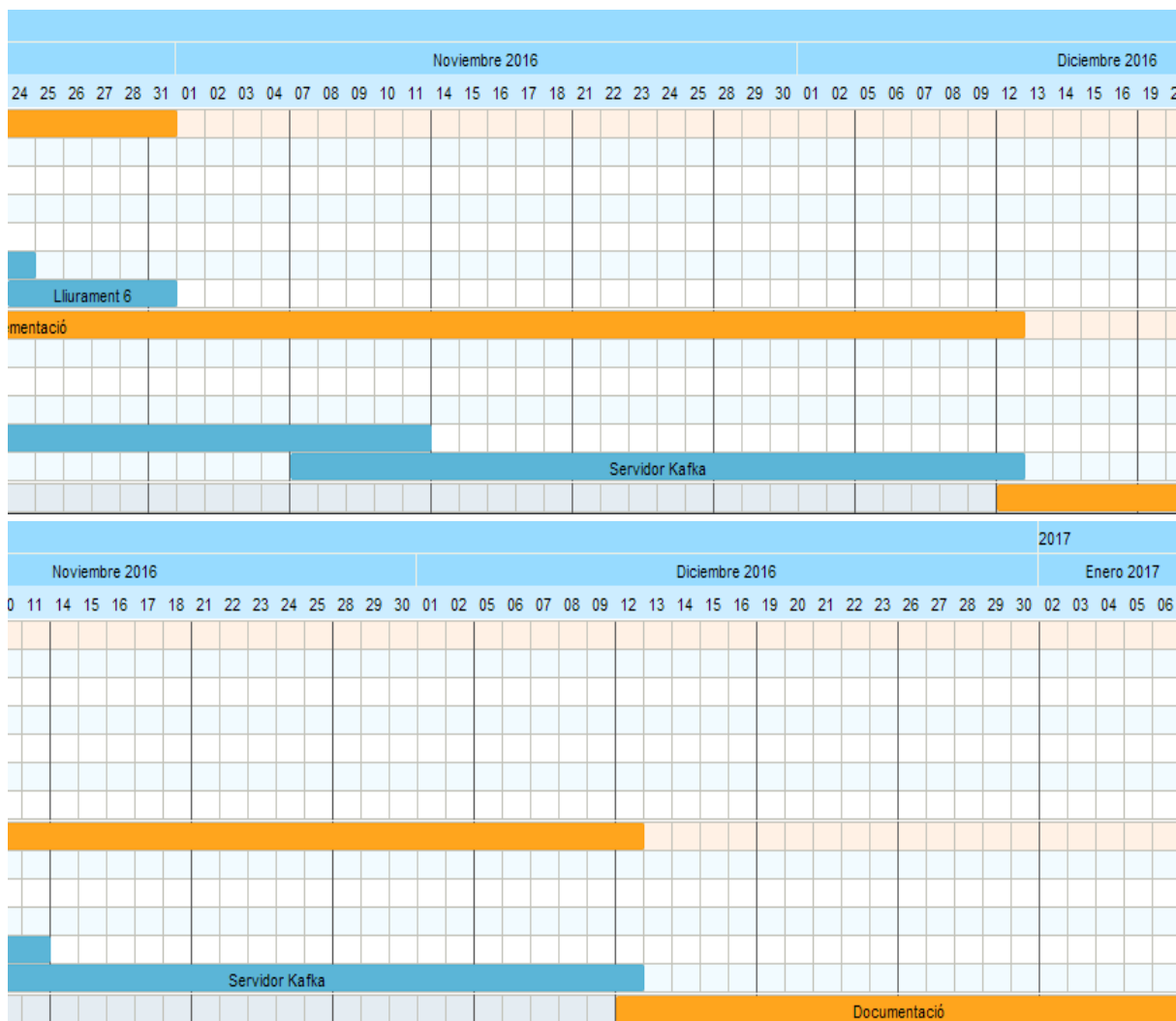
En el projecte, puc dir que hi han hagut poques desviacions, ja que només puc destacar una desviació en tot el treball i ha sigut al final d'aquest quan estava document i realitzant la última part del servidor Kafka.

El problema que vaig trobar va ser que al intentar implementar la connexió entre el servidor RESTful i el servidor Kafka vaig tenir molts problemes amb les llibreries necessàries i no trobava cap tipus de solució, fins que vaig pensar la solució imaginativa que tenim ara. Per culpa d'aquesta desviació la última part d'implementació va trigar dues setmanes més de les esperades, perquè també va coincidir amb un període de vacances. Per sort, vaig poder solucionar-ho i em vaig posar a documentar just al acabar d'implementar, ja que quan vaig tenir el problema no podia concentrar-me en documentar si tenia l'altre problema. Per tant, la documentació va ser retardada dues setmanes, fent que el període de revisió del document d'entrega sigui menor.

També he trobat que durant el mes de desembre, al haver bastants períodes festius i que durant els dies laborables havia de treballar en un projecte aliè a aquest, ha fet dificultós el poder treballar còmodament i ha fet que les hores dedicades a treball diari baixessin. Però a la planificació ja ho vaig tenir en compte, i es per aquesta raó que vam deixar un gran espai temporal per la documentació, ja que aquesta seria durant el període festiu d'hivern.

A continuació tenim un diagrama de Gant per poder veure les desviacions més fàcilment, aquest ha sigut dividit per a una millor visualització.





13. Anàlisi de costos

a. Consideracions

Una vegada establerts els recursos necessaris pel projecte, necessito posar-li un cost a aquests. Per això realitzaré un pressupost el més detallat possible.

En el pressupost no tindrè en compte si el costos varien durant l'execució del projecte, degut a que el projecte és de duració curta. Tampoc afegiré un marge de benefici ja que el projecte no és un projecte amb ànim de lucre.

b. Recursos utilitzats

Per la realització del projecte serà necessària la utilització d'uns determinats recursos, tant de hardware com de software.

Hardware

El hardware utilitzat per aquest projecte serà:

- HP Pavillion dv6 Notebook PC
- Samsung Core Prime
- Servidor Apache Tomcat v.8
- Servidor Apache Kafka

Software

El software utilitzat per aquest projecte serà:

- Microsoft Windows 7
- Microsoft Office 2013
- Adobe Reader X
- Eclipse
- Android Studio
- Google Cloud Messaging
- Git
- Postman

Humans

Els recursos humans utilitzats només ha sigut una persona, un estudiant que està realitzant el projecte de fi de grau. Amb ajuda d'un professor de la UPC i encarregat del projecte SUPERSEDE, amb el qual realitza reunions setmanals per verificar el progrés del treball, intentar solucionar dubtes i assegurar-se de que es compleixen els requeriments del projecte.

c. Pressupost dels recursos humans

Per realitzar el pressupost dels recursos humans agafaré els salaris mínims proposats per la web PagePersonnel [9]. Encara que realment, tots els rols els fa la mateixa persona i aquesta és un estudiant.

<i>Rol</i>	<i>Hores estimades</i>	<i>Preu per hora</i>	<i>Preu total</i>
<i>Cap del projecte</i>	120h	18,22€/h	2187,5€
<i>Desenvolupador</i>	300h	9,375€/h	2812,5€
<i>Responsable de les proves</i>	85h	11,45€/h	973,95€
<i>Total estimat</i>	505h		5973,95€

d. Pressupost pel hardware

En la part del pressupost del hardware establiré els costos del hardware pel nostre projecte a partir de la seva vida útil i l'amortització de cada element.

<i>Producte</i>	<i>Preu</i>	<i>Vida útil</i>	<i>Cost estimat</i>
<i>HP Pavillion dv6 Notebook PC</i>	1100€	5 anys	110€
<i>Samsung Core Prime</i>	160€	3 anys	26,67€
<i>Servidor Apache Tomcat v.8</i>	2800€	N/A	2800€
<i>Servidor Apache Kafka</i>	2800€	N/A	2800€
<i>Total estimat</i>			5726,67€

Com podem veure, només he afegit un servidor Apache Tomcat, en comptes de dos, ja que per les proves i el volum de dades necessari, amb la opció que em proporciona el servidor Apache Tomcat de poder tenir dos projectes alhora, ja puc realitzar el treball que necessitava sense necessitat d'afegir un servidor addicional.

e. Pressupost pel software

La majoria de software que utilitzo és gratuït però hi ha algun que és de pagament.

Producte	Preu	Cost estimat
<i>Microsoft Windows 7 Home Premium</i>	Ve inclòs a l'ordinador	0€
<i>Microsoft Office 2013 Professional</i>	7€/mes	28€
<i>Adobe Reader X</i>	Gratuït	0€
<i>Eclipse</i>	Gratuït	0€
<i>Android Studio</i>	Gratuït	0€
<i>Google Cloud Messaging</i>	Gratuït	0€
<i>Git</i>	Gratuït	0€
<i>Postman</i>	Gratuït	0€
Total estimat		28€

f. Contingència

La contingència en el pressupost serà d'un 10% del total fins ara. Hem posat només un 10% perquè crec que més percentatge seria excessiu per la petita grandària del projecte.

Costos	Cost estimat
<i>Recursos Humans</i>	5973,95€
<i>Hardware</i>	5726,67€
<i>Software</i>	28€
<i>Subtotal</i>	11728,62€
Contingència estimada(10%)	1172,86€

g. Imprevistos

Com he parlat en la secció anterior de planificació temporal, els dos gran imprevistos que podria trobar-me era durant la realització de la implementació i/o la fase de documentació. Com explico en aquella secció, he deixat un parell de setmanes lliures per poder utilitzar-les si es cau. Sabent això, podria fer una petita valoració sobre els costos dels imprevistos, en particular sobre el cost dels recursos humans ja que la resta no variaria el seu cost.

Rol	Hores estimades	Preu per hora	Preu total
<i>Cap del projecte</i>	5h	18,22€/h	91,1€
<i>Desenvolupador</i>	20h	9,375€/h	187,5€
<i>Responsable de les proves</i>	5h	11,45€/h	57,25€
Total estimat	30h		335,85€

h. Control de gestió

Per controlar que el pressupost no es desviï del que he pactat anteriorment, realitzaré control periòdics. Aquests controls seran per la revisió de la previsió dels recursos humans, ja que l'altre part, tant hardware com a software, és una inversió inicial que no variarà en tot el projecte.

Tindrè dos tipus de controls periòdics, els primers seran setmanals durant la reunió amb el director del treball. Repassarem com a anat la setmana i intentarem fer una revisió si tot va en la direcció correcta. L'altre tipus serà una vegada acabada una de les fases fer una anàlisi global per veure tot el conjunt fet i veure més en detall si hi hagut alguna desviació que amb les reunions setmanals no s'ha pogut apreciar.

En cas de tenir una desviació en el nostre projecte, hauré d'utilitzar la part del pressupost de contingència que és limitat.

Les desviacions i les fórmules per calcular-les són les següents:

- Desviament de mà d'obra en preu=(cost estimat – cost real)*consum hores real
- Desviament de mà d'obra en consum=(consum hores estimades – consum hores real)*cost estimat
- Desviament total de mà d'obra=total cost estimat mà d'obra – total cost real mà d'obra

i. Resum

Agafant tots els costos anterior faré una valoració global del cost total del pressupost.

Concepte	Cost
<i>Recursos humans</i>	5973,95€
<i>Hardware</i>	5726,67€
<i>Software</i>	28€
<i>Contingència</i>	1172,86€
<i>Imprevistos</i>	335,85€
Total estimat	13237,33€

14. Sostenibilitat i Compromís social

a. Econòmica

El projecte és sostenible econòmicament ja que té una avaluació aproximada dels costos dels recursos humans i materials, fent possible una planificació. Aquesta avaluació la podem veure en la secció anterior, el pressupost. L'únic inconvenient és que el pressupost només considera la part de desenvolupament del projecte i la part de la seva vida útil, per tant no es tenen en compte possibles costos de manteniment i/o millora.

A més el projecte, al ser el treball de fi de grau fa que el cost de la mà d'obra sigui de les més econòmiques, ja que és un estudiant qui ho fa. Per tant els costos d'aquest són pràcticament mínims. A més la majoria de recursos, tant de software com de hardware, ja es tenien i el fet de que sigui un estudiant qui realitzi el projecte, fa que el temps dedicat per una tasca sigui acurat, ja que té un temps limitat per realitzar el projecte, per tant l'establiment de prioritats en la realització de tasques és molt important.

Per últim aquest projecte és sostenible econòmicament perquè forma part d'un projecte més gran. Concretament, el nostre projecte és la primera part d'aquest, com ja he explicat en seccions anterior, per tant és important que sigui factible de realitzar, ja que tindrà una aplicació real.

b. Social

Actualment que l'estat espanyol estigui en mig d'una crisi econòmica i social, fa que el projecte es vegi una mica afectat per aquests factor, però un punt a favor és que el sector al que va dirigit, el de telecomunicacions i informàtica, està creixent. A més, han sigut les pròpies empreses del sector, les que han demanat la realització d'aquest projecte per poder millorar les seves aplicacions. Es per això, que els casos d'ús del projecte seran sobre les aplicacions d'aquestes empreses. Per tant, es pot dir que hi ha una necessitat i un desig per part del sector per intentar progressar amb les seves aplicacions i poder extreure'ls el màxim rendiment fent que es generin llocs de treball gràcies a la creacions d'aquests projectes.

c. Ambiental

L'impacte ambiental del projecte és mínim ja que gairebé tot els recursos necessaris, tant software com a hardware, ja estan disponibles sense necessitat de comprar ni construir res. L'únic impacte o consum que té és l'electricitat que consumeixen els elements, l'ordinador i els servidors. En el cas d'aquests últims, les proves es realitzen al ordinador així que no malgastem electricitat amb els servidors. A més, el projecte al ser modular he pogut implementar sense la necessitat de tenir tot el sistema encès a la vegada, sinó la part que necessitàvem en cada moment.

Una altre aspecte a favor és que el projecte tindria pràcticament el mateix impacte ambiental si l'hagués fet una altre persona que no fos un estudiant de TFG, per tant té un impacte ambiental constant.

d. Resum

Amb tota la informació anterior puc establir la matriu de sostenibilitat proposada a la guia de l'assignatura.

Sostenibilitat	Econòmica	Social	Ambiental
<i>Planificació</i>	Viabilitat econòmica	Millora en la qualitat de vida	Analisi de recursos
<i>Valoració Resultats</i>	8	6	9
	Cost final vs previsió	Impacte en l'entorn social	Consum de recursos
<i>Valoració Riscos</i>	10	9	10
	Adaptació a canvis d'escenaris	Danys socials	Danys ambientals
<i>Valoració</i>	2	0	0
Total			54

15. Justificació de les competències tècniques

a. Coneixement d'assignatures d'especialitats

Durant el període de la especialitat, he realitzat diferents assignatures que m'han ajudat a establir un certs coneixements en determinades matèries. Algunes d'aquestes assignatures, m'han ajudat a l'hora de realitzar el projecte en diferents aspectes que passaré a explicar. Aquests coneixements adquirits són:

- Coneixements en RESTful APIs, el qual m'ha permès fer la part del servidor que envia la configuració a la aplicació mòbil.
- Coneixements sobre l'arquitectura del software, l'aplicació dels diferents patrons i la correctesa del codi.
- Coneixements previs dels llenguatges utilitzats en el projecte. En diverses assignatures impartides en la especialitat he utilitzat els dos llenguatges que utilitzo en el projecte, Android i Java.
- Metodologies de treball per poder obtenir el màxim rendiment en el projecte, per exemple el coneixement de la metodologia de treball Agile.

b. Característiques de l'especialitat

El projecte proposat s'adequa a les característiques principals de l'especialitat d'Enginyeria de Software.

- La primera característica important és que el projecte està realitzat de manera que satisfà els requisits demanats pels usuaris de la nostra aplicació. A més d'assegurar que el seu comportament és fiable i eficient i el seu manteniment és assequible, gràcies a l'aplicació els diferents mètodes i principis d'Enginyeria del Software. Ja que la nostre aplicació realitza tot el que demana el client amb la màxima claredat i correctesa possible.
- Un altra característica també relacionada amb la primera és que a partir dels objectius que vull aconseguir i els requisits de software necessaris per satisfer les necessitats dels clients saber valorar les diferents limitacions derivades del cost, temps i de l'existència d'altres sistemes semblants ja desenvolupats. Per exemple, com ja he esmentat anteriorment, el projecte té un temps limitat, el que he intentat és reaprofitar codi d'altres organitzacions per haver de refer un codi que ja està fet. Més concretament he utilitzat exemples de Google per poder una connexió entre servidors i aplicació més ràpida i eficient en el límit de temps que disposem.
- L'última principal característica que compleix el nostre projecte és la capacitat que he tingut en l'anàlisi i identificació dels problemes. Una de les activitats que ha ajudat a garantir aquesta característica han sigut les reunions setmanals amb els professors involucrats en el

projecte de manera que em permetia poder dissenyar, desenvolupar, implementar i verificar les nostres solucions de software per assegurar-me que estava utilitzant les teories, models i tècniques adients per realitzar el treball.

c. Competències tècniques

Les competències tècniques en el nostre projecte són:

- CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [En profunditat]. Aquesta competència ha sigut escollida degut a que el projecte estarà dins d'un altre projecte més gran, per tant la nostra aplicació serà part d'un sistema més gran i haurà de funcionar perfectament per poder assegurar el flux d'informació en el sistema.
- CES1.4: Desenvolupar, mantenir i avaluar serveis i aplicacions distribuïdes amb suport de xarxa. [Bastant]. Com he comentat en altres lliuraments la nostra aplicació és basa en dos servidors i una aplicació per a mòbils Android. La comunicació entre aquests dispositius és a través de la xarxa, sense aquesta comunicació la nostra aplicació no tindria cap sentit.
- CES1.7: Controlar la qualitat i dissenyar proves en la producció de software. [En profunditat]. Aquesta competència s'ha escollit ja que, com he dit anteriorment, l'aplicació va integrada en una aplicació més gran, per tant he d'assegurar-me que funciona correctament l'aplicació independent dels altres elements per evitar errors majors en el sistema. Per tant he de comprovar i realitzar proves pel seu correcte funcionament.
- CES2.1: Definir i gestionar els requisits d'un sistema software. [Bastant]. El projecte té un requisits clars i ben definits degut a que és projecte el qual ja existia i tenien una necessitat per tant es important saber gestionar tots els requisits d'aquest sistema per assegurar-nos la qualitat d'aquest.
- CES2.2: Dissenyar solucions apropiades en un o més dominis d'aplicació, utilitzant mètodes d'enginyeria del software que integrin aspectes ètics, socials, legals i econòmics. [Bastant]. Aquesta última competència és important per el fet que l'aplicació, com he reiterat anteriorment, està dins d'una aplicació més gran, per tant no sé que faran després que el meu projecte estigui conclòs. Per tant, he d'assegurar-me d'utilitzar totes les tècniques i mètodes al nostre abast per fer la millor solució possible tant per el correcte i eficient funcionament, com per que si hagués una possible millora, canvi o manteniment del nostre sistema, aquesta sigui fàcil i ràpida de realitzar.

16. Conclusions

a. Assoliment d'objectius

Per concloure el projecte, he d'esmentar que els dos objectius presentats han sigut complerts.

L'objectiu número 1, ha sigut aconseguit amb la realització de la aplicació mòbil per al sistema operatiu Android, que permet la monitorització de les diferents mètriques. Aquestes mètriques són escollides mitjançant l'enviament d'un document de configuració a l'aplicació mòbil. Més tard, les mètriques demanades són obtingudes i enviades a un servidor per ser emmagatzemades.

L'objectiu número 2, ha sigut dut a terme gràcies a la creació de dos servidors RESTful Service. Un servidor és l'encarregat d'enviar la configuració a la aplicació mòbil i registrar els dispositius mòbils i l'altre, és l'encarregat de emmagatzemar la informació que prové dels dispositius mòbils a un servidor Kafka. Aquesta estructura ens ha permès donar suport a l'aplicació perquè pugui funcionar a un nivell òptim

A més, s'ha complert la planificació establerta, tot i la desviació apareguda al final de la fase de implementació. També remarcar, que juntament amb els objectius, s'ha complert els requisits funcionals i no funcionals perquè el sistema funcioni de la manera més òptima possible.

b. Treball futur

Encara que he realitzat tots els objectius de la manera desitjada, com tots els projectes, és millorable. Per tant, m'agradaria donar unes quantes idees per projectes futurs de millores al sistema.

- Adaptar el codi dels servidors actuals perquè tinguin una millor resposta envers a una alta quantitat de dades.
- Buscar més mètriques per la monitorització, per tal de donar més eines per un millor anàlisi.
- Adaptar l'aplicació mòbil per a altres sistemes operatius, com per exemple IOS.
- Realitzar millores en les comunicacions entre els elements, per exemple fer que les crides siguin asíncrones.
- Crear una llibreria perquè les aplicacions mòbils no hagin d'enviar les dades a un servidor RESTful Apache Tomcat, que es l'encarregat d'enviar les dades al servidor Apache Kafka, sinó que ho enviïn directament a aquest.

c. Valoració personal

Aquest projecte ha sigut molt didàctic ja que m'ha ajudat a millorar en el meu desenvolupament com a informàtic, ja que he repassat les diferents tècniques que m'han ensenyat durant el Grau i a après de noves que encara no sabia. A més a més, m'ha permès conèixer de més a prop com funcionen els projectes que desenvolupa la universitat i conèixer gent molt brillant que treballa en el projecte com és el meu director, Xavier Franch Gutiérrez i el meu codirector, Marc Oriol Hilari.

Gràcies al projecte, m'ha ajudat a revalidar la idea de que un projecte no és d'una sola persona sinó d'un equip. Ja que hi hagut vegades durant la realització del projecte, que no sabia com trobar una solució i gràcies als directors del treball hem pogut trobar-ne una que s'adequava al que necessitàvem.

Crec que és important que el treball de fi de grau sigui un aspecte important de la teva educació universitària i no un tràmit més del Grau i això he intentat aconseguir, amb l'ajuda dels directors i, segons la meua opinió diria que ho he aconseguit.

17. References

- [1] S. BENBERNOU, L. CAVALLARO, M. SAID HACID, R. KAZHAMIKIN, G. KECSKEMETI, J. POIZAT, F. SILVESTRI, M. UHLIG, AND B. WETZSTEIN. STATE OF THE ART REPORT, GAP ANALYSIS OF KNOWLEDGE ON PRINCIPLES, TECHNIQUES AND METHODOLOGIES FOR MONITORING AND ADAPTATION OF SBAs. DELIVERABLE # PO-JRA-1.2.1, 2008.
- [2] A. BAUER, M. LEUCKER, AND C. SCHALLHART. RUNTIME VERIFICATION FOR LTL AND TLTL. ACM TRANS. SOFTW. ENG. METHODOL., 20(4):14, 2011.
- [3] R. ATTERER, M. WNUK, AND A. SCHMIDT. KNOWING THE USER'S EVERY MOVE: USER ACTIVITY TRACKING FOR WEBSITE USABILITY EVALUATION AND IMPLICIT INTERACTION. IN PROCEEDINGS OF INTERNATIONAL CONFERENCE ON WORLD WIDE WEB (WWW), 2006.
- [4] A. GILLIES. SOFTWARE QUALITY: THEORY AND MANAGEMENT. 3RD EDITION, EDITORIAL LULU.COM.
- [5] L. ATZORI, A. IERA, AND G. MORABITO. THE INTERNET OF THINGS: A SURVEY. INTERNATIONAL JOURNAL OF COMPUTER AND TELECOMMUNICATIONS NETWORKING, 54(15):2787-2805.
- [6] R. ALI, C. SOLÍS, I. OMORONYIA, M. SALEHIE, AND B. NUSEIBEH. SOCIAL ADAPTATION - WHEN SOFTWARE GIVES USERS A VOICE. IN J. FILIPE AND L. A. MACIASZEK, EDITORS, ENASE, PAGES 75-84. SciTEPRESS, 2012.
- [7] E.R. NÚÑEZ-VALDÉZ, J.M. CUEVA LOVELLE, O. SANJUÁN MARTÍNEZ, V. GARCÍA-DÍAZ, P. ORDÓÑEZ DE PABLOS, AND C.E. MONTENEGRO MARÍN. IMPLICIT FEEDBACK TECHNIQUES ON RECOMMENDER SYSTEMS APPLIED TO ELECTRONIC BOOKS. COMPUTERS IN HUMAN BEHAVIOR, 28(4):1186-1193, 2012.
- [8] D. PAGANO AND B. BRUEGGE. USER INVOLVEMENT IN SOFTWARE EVOLUTION PRACTICE: A CASE STUDY. IN NOTKIN ET AL. [NCP13], 953–962.
- [9] PAGE PERSONNEL (2016). ESTUDIOS DE REMUNERACIÓN 2016 – TECNOLOGÍA [EN LÍNEA]. DISPONIBLE A:
[HTTP://WWW.PAGEPERSONNEL.ES/SITES/PAGEPERSONNEL.ES/FILES/ER_TECNOLOGIA16.PDF](http://www.pagepersonnel.es/sites/pagepersonnel.es/files/er_tecnologia16.pdf)
- [10] CLOUD MESSAGING(2016). GOOGLE. DISPONIBLE A:
[HTTPS://DEVELOPERS.GOOGLE.COM/CLOUD-MESSAGING/](https://developers.google.com/cloud-messaging/)
- [10] CLOUD MESSAGING CLIENT(2016). GOOGLE. DISPONIBLE A:
INTERNET:HTTPS://DEVELOPERS.GOOGLE.COM/CLOUD-MESSAGING/ANDROID/CLIENT
- [11]RETROFIT(2016) DISPONIBLE A: [HTTPS://SQUARE.GITHUB.IO/RETROFIT/](https://square.github.io/retrofit/)
- [12] POSTMAN(2016) DISPONIBLE A: [HTTPS://WWW.GETPOSTMAN.COM/](https://www.getpostman.com/)

- [13] SPROTT, D. , WILKES, L. — UNDERSTANDING SERVICE-ORIENTED ARCHITECTURE. DISPONIBLE A: [HTTPS://MSDN.MICROSOFT.COM/EN-US/LIBRARY/AA480021.ASPX](https://msdn.microsoft.com/en-us/library/aa480021.aspx)
- [14] DOUGLAS, B. — SERVICE ARCHITECTURE. DISPONIBLE A: [HTTP://WWW.SERVICE-ARCHITECTURE.COM/ARTICLES/WEB-SERVICES/SERVICE-ORIENTED_ARCHITECTURE_SOA_DEFINITION.HTML](http://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html)
- [15] LEE, Y., IYENGAR, S. S., MIN, C., JU, Y., KANG, S., PARK, T., LEE, J., RHEE, Y. SONG, J.: MOBICON: A MOBILE CONTEXT-MONITORING PLATFORM. COMMUN. ACM. 55, 54-65 (2012)
- [16] BETTINI, C., MAGGIORINI, D., RIBONI, D.: DISTRIBUTED CONTEXT MONITORING FOR THE ADAPTATION OF CONTINUOUS SERVICES. WWW. 10, 503-528 (2007)
- [17] KANG, S., LEE, Y., MIN, C., JU, Y., PARK, T., LEE, J., RHEE, Y., SONG, J.: ORCHESTRATOR: AN ACTIVE RESOURCE ORCHESTRATION FRAMEWORK FOR MOBILE CONTEXT MONITORING IN SENSOR-RICH MOBILE ENVIRONMENTS. IN: IEEE PERCOM, PP. 135-144 (2010)
- [18] KANG, S., LEE, J., JANG, H., LEE, Y., PARK, S., SONG, J.: A SCALABLE AND ENERGY-EFFICIENT CONTEXT MONITORING FRAMEWORK FOR MOBILE PERSONAL SENSOR NETWORKS. IEEE TRANS. MOBILE COMPUT. 9, 686-702 (2010)
- [19] RIVA, O.: CONTORY: A MIDDLEWARE FOR THE PROVISIONING OF CONTEXT INFORMATION ON SMART PHONES. IN: MIDDLEWARE, PP. 219-239 (2006)
- [20] MARC ORIOL, JORDI MARCO, OSCAR CABRERA, QUIM MOTGER, FARNAZ FOTROUSI, DENISSE MUÑANTE, ANNA PERINI. D1.4: COMPREHENSIVE MONITORING TECHNIQUES, v1". PUBLIC DELIVERABLE OF PROJECT SUPERSEDE. 2016